A Reproduced Copy

OF

---

N71-75077

Reproduced for NASA

*by the*

**NASA Scientific and Technical Information Facility**

FFNo 672 Aug 65

OPTIMUM PACKAGING DESIGN AND COMPONENT SPARING POLICIES FOR....

D. E. Wagner, 1968

*Published on demand by*
UNIVERSITY MICROFILMS
*University Microfilms Limited, High Wycombe, England*
*A Xerox Company, Ann Arbor, Michigan, U.S.A.*

N71-75077
(ACCESSION NUMBER) (THRU)
173 None
TMX-67300 (CODE)
(NASA CR OR TMX OR AD NUMBER) (CATEGORY)
FACILITY FORM 602

* * *

This is an authorized facsimile and was produced by microfilm-xerography in 1970 by University Microfilms, A Xerox Company, Ann Arbor, Michigan, U.S.A.

* * *

8800

This dissertation has been microfilmed exactly as received

69-8513

WAGNER, Donald Edward, 1934-
OPTIMUM PACKAGING DESIGN AND COMPONENT SPARING POLICIES FOR EXTENDED MANNED SPACE MISSIONS.

Texas A and M University, Ph.D., 1968
Engineering, industrial

University Microfilms, Inc., Ann Arbor, Michigan

0-8581-B

# OPTIMUM PACKAGING DESIGN AND COMPONENT SPARING POLICIES FOR EXTENDED MANNED SPACE MISSIONS

A Dissertation

By

DONALD EDWARD WAGNER

Submitted to the Graduate College of the
Texas A&M University in
partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 1968

Major Subject: Industrial Engineering

# OPTIMUM PACKAGING DESIGN AND COMPONENT SPARING POLICIES FOR EXTENDED MANNED SPACE MISSIONS

A Dissertation

By

DONALD EDWARD WAGNER

Approved as to style and content by:

(Chairman of Committee)

(Head of Department)

R.R. Hocking
(Member)

(Member)

(Member)

(Member)

August 1968

## ABSTRACT

Optimum Packaging Design and Component Sparing Policies for Extended Manned Space Missions. (August 1968)

Donald E. Wagner, B. S., Lamar State College of Technology; M. S., Northwestern University;

Directed by: Dr. Glen Self

The primary objective of this research is to provide an analytical solution to the problem of optimizing the design of manned spacecraft systems with respect to the modular level at which on-board replacement of failed modules should be carried out. Failed modules are assumed to be non-repairable. The optimum design is considered as a system with the minimum weight, consistent with mission requirements, for which a required reliability can be attained. A specific set of design alternatives in the form of a hierarchy of candidate replacement modules is used as the framework from which an optimum design is selected.

The reliability expressions for spared systems are derived under the assumption that the components follow a negative exponential failure distribution. A detailed derivation of the reliability of a series-parallel system with spares is presented, and comparisons are made with a series spared system.

The effect of having common components in spared systems is discussed and illustrated in graphs for specific cases. An advantage of increasing component weight to achieve commonality of components is demonstrated.

The optimum module level problem is solved using a dynamic programming approach for the case where each module is unique. For the case with common modules an integer programming model is formulated. A solution is obtained by a specialized algorithm of the branch and bound type which uses dynamic programming as a bounding method. The attendant assumptions, considerations, and data requirements for the model are outlined.

Some assumptions are (1) that there is no degradation of the reliability of spares while not in use, (2) each of the modules in the basic system are required for the successful completion of the mission, and (3) the penalties for sparing a given module, such as fault isolation and replacement times, can be expressed in terms of a common parameter.

An example problem is presented and used as a basis for discussion. The solution is illustrated by a tree of feasible solutions.

## ACKNOWLEDGEMENTS

The primary topic of this research was originated by my supervisor, Mr. W. E. Stoney, Jr., Chief, Advanced Spacecraft Technology Division, NASA/Manned Spacecraft Center. He expressed the idea by a sketch of various sized "black boxes" within which were drawn other "black boxes" representing the components of the larger modules of the spacecraft. The question was then posed as: "Which 'black boxes' should we carry as spares?" From this preliminary discussion the research documented herein has evolved. I would like to express my appreciation to Mr. Stoney both for providing my research subject and for his sponsorship through the NASA Graduate Study Program.

Dr. Glen Self, Associate Professor of Industrial Engineering, was present both at the original meeting and several later ones to discuss my research topic. I wish to express my sincere thanks to Dr. Self for his invaluable guidance, suggestions, and ideas throughout the course of this work.

To Dr. A. W. Wortham, Head of Industrial Engineering, I would like to express my appreciation both for his assistance and counseling throughout my graduate program as well as for his encouraging me to return to graduate study. It was largely through Dr. Wortham's efforts that

a Ph.D. program in Industrial Engineering was initiated at Texas A&M University.

I wish to express my gratitude to Dr. W. L. Meier, Associate Professor of Industrial Engineering, Dr. R. R. Hocking, Professor of Statistics, and Dr. D. R. Drew, Head-Design and Traffic Department, for their time and efforts spent on my behalf as my graduate committee members.

My appreciation is also expressed to many friends and colleagues for their assistance and suggestions during the formative stages of the work.

I would like to thank Mr. and Mrs. Paul Bradley, for their diligence and patience in producing the manuscript from my handwritten notes.

The list of acknowledgements would be incomplete without mention of the understanding and forbearance of my wife and children during the course of my graduate program.

TABLE OF CONTENTS


| | | Page |
|---|---|---|
| ABSTRACT | | iii |
| ACKNOWLEDGEMENTS | | v |
| LIST OF TABLES | | ix |
| LIST OF FIGURES | | x |
| Chapter | | |
| I. | INTRODUCTION | 1 |
| | Objectives | 3 |
| | Statement of the Problem | 4 |
| | General Assumptions | 5 |
| II. | REVIEW OF THE LITERATURE | 8 |
| | Redundancy Optimization Problems | 9 |
| | Applications of Redundancy Optimization to Space Systems | 17 |
| | Optimum Module Size | 30 |
| | Branch and Bound Solution Methods | 34 |
| | Summary | 43 |
| III. | DEVELOPMENT OF THE RELIABILITY OF REDUNDANT SYSTEMS | 46 |
| | Time Independent | 47 |
| | Exponential Failure Law with Spares | 48 |
| | Parallel Components with Spares | 50 |
| | Comparison of Reliability Values for Spared Systems | 60 |
| | Effects of Commonality | 68 |
| | Design Generalizations | 72 |
| IV. | FORMULATION OF THE OPTIMUM MODULE SIZE PROBLEM | 77 |
| | Assumptions and Considerations | 85 |
| | Approaches to the Model | 91 |
| | Integer Programming Formulation | 99 |
| | Data Requirements | 103 |

| Chapter | | Page |
|---|---|---|
| V. | A SOLUTION PROCEDURE FOR THE OPTIMUM MODULE SIZE | 105 |
| | Discussion of the Solution Approach | 105 |
| | A Specific Algorithm | 115 |
| | Theoretical Development | 120 |
| | Variants of the Algorithm | 122 |
| | Sensitivity Analysis | 126 |
| | An Example Problem | 128 |
| VI. | SUMMARY AND CONCLUSIONS | 137 |
| VII. | RECOMMENDATIONS FOR FUTURE RESEARCH | 142 |
| APPENDIX | | |
| | Reliability Versus Weight Curves for Example Problem | 146 |
| REFERENCES | | 154 |
| VITA | | 160 |

# LIST OF TABLES


| TABLE | | PAGE |
|---|---|---|
| 3.1 | TRANSITION MATRIX FOR PARALLEL SPARED SYSTEM . . . . . | 52 |
| 4.1 | VALUES OF $k_{ij}^{(p)}$ FOR FIGURE 4.1 . . . . . | 82 |
| 4.2 | DYNAMIC PROGRAMMING STAGES FOR THE EXAMPLE PROBLEM OF FIGURE 4.2 . . . . . | 94 |
| 5.1 | VALUES OF PARAMETERS FOR EXAMPLE PROBLEM . . | 129 |

## LIST OF FIGURES


| FIGURE | | PAGE |
|---|---|---|
| 2.1 | Illustration of a series-parallel system . . | 14 |
| 2.2 | Cell configuration . . . . . | 19 |
| 2.3 | Commonality configuration . . . . . | 21 |
| 2.4 | Possible consumption of two spares with three cells . . . . . | 22 |
| 2.5 | Example of a solution tree . . . . . | 39 |
| 3.1 | Parallel system with spares, N=2, m=3 . . . | 52 |
| 3.2 | Representation of component sets of Figures 3.3 to 3.7 . . . . . | 61 |
| 3.3 | Reliability vs $\lambda t$ for series-parallel and series systems with spares, illustrated in Figure 3.2, N=1 . . . . . | 63 |
| 3.4 | Reliability vs $\lambda t$ for series-parallel and series systems with spares, illustrated in Figure 3.2, N=2 . . . . . | 64 |
| 3.5 | Reliability vs $\lambda t$ for series-parallel and series systems with spares, illustrated in Figure 3.2, N=3 . . . . . | 65 |
| 3.6 | Reliability vs $\lambda t$ for series-parallel and series systems with spares, illustrated in Figure 3.2, N=5 . . . . . | 66 |
| 3.7 | Reliability vs $\lambda t$ for series-parallel and series systems with spares, illustrated in Figure 3.2, N=10 . . . . . | 67 |
| 3.8 | Effect of commonality on spares required for various numbers of components in system, $\lambda t$=.10, $R \geq .99$ . . . . . | 69 |
| 3.9 | Effect of commonality on spares required with module weight increased 50% for commonality case, $\lambda t$=.10, $R \geq .99$ . . . . . | 71 |

| FIGURE | | PAGE |
|---|---|---|
| 3.10 | Representation of a system with n parallel components and m spares . . . . . | 73 |
| 3.11 | Reliability vs number of components in parallel, $\lambda t=.5$ . . . . . | 74 |
| 3.12 | Reliability vs number of components in parallel, $\lambda t=1.0$ . . . . . | 75 |
| 4.1 | Example system of candidate modules . . . . | 78 |
| 4.2 | Example system of candidate modules without commonality . . . . . | 95 |
| 5.1 | Solution tree for feasible values of $\delta_{ij}^{(p)}$ for example problem of Figure 4.1 . . . . . | 107 |
| 5.2 | Reliability vs weight for a module and its constituent components . . . . . | 110 |
| 5.3 | Solution tree for feasible values of $\delta_{ij}^{(p)}$ for example problem of Figure 4.1 indicating solutions excluded . . . . . | 130 |
| A.1 | Reliability vs weight for module (i,j,p)= (1,2,1) and its constituent components . . . | 147 |
| A.2 | Reliability vs weight for module (i,j,p)= (2,2,1) and its constituent components . . . | 148 |
| A.3 | Reliability vs weight for module (i,j,p)= (3,2,1) and its constituent components . . . | 149 |
| A.4 | Reliability vs weight for module (i,j,p)= (1,2,2) and its constituent components . . . | 150 |
| A.5 | Reliability vs weight for module (i,j,p)= (4,2,2) and its constituent components . . . | 151 |
| A.6 | Reliability vs weight for module (i,j,p)= (1,3,1) and its constituent components . . . | 152 |
| A.7 | Reliability vs weight for module (i,j,p)= (2,3,1) and its constituent components . . . | 153 |

# CHAPTER I

## INTRODUCTION

Until the present time, manned space missions have been of sufficiently short duration to allow achievement of desired mission reliability without requiring planned maintenance. However, recent studies [11,43,50]† have indicated that for the long space missions being planned for the future, such as a Mars mission or Earth orbital flights of greater than six months duration, some type of maintenance by the crew will be required.

It is yet to be determined to what extent maintenance should be performed during a space mission. There are many degrees of maintenance to consider for correcting malfunctions, ranging from merely automatic switching of redundant components into active use to the complicated maintenance functions implied by bench repair. The question of degrees of maintenance to consider, while pertinent to the overall subject of design of spacecraft for long space missions, has been discussed qualitatively in several publications [8,11,42,43,50] and is not explicitly approached in this research. Several studies [42,43] have indicated the desirability of a maintenance policy which involves repair

†The citations on the following pages follow the style of the Journal of the Operations Research Society of America.

of malfunctions solely by replacement of defective items. This research represents the development of quantitative assessment of the reliability which is achieved by such an approach as well as optimization of the spacecraft configuration with respect to carrying out a policy of limiting the maintenance actions of the crew primarily to replacement of failed components with spares.

Since there has been practically no experience with maintainable systems in space, there are little or no actual data upon which to base an analysis of the type represented by this research. Therefore, the procedures developed have been generalized to accept future data and to indicate some of the data required for quantitative assessment of design alternatives in designing maintainable space systems.

In the transition from relatively short space missions to durations of over a year in space, there is a need to consider new concepts of spacecraft design. One method of probing for the most promising of several alternatives is to use preliminary design data as the input to a rigorous decision tool such as a mathematical model and observe the indicated results. By varying the input data, the effects on the output can be analyzed, and the model further refined. This research is an effort to develop a basic tool for analyzing alternate designs for

maintainable spacecraft; as such, it is expected that, through its use with estimated data, extensions to and revisions of the model will result.

## Objectives

The primary objective of this research is to provide an analytical solution to the problem of optimizing the design of manned spacecraft systems with respect to the modular level at which on-board replacement of failed modules should be carried out. The optimum design will be considered as a system with the minimum weight, consistent with mission requirements, for which a required reliability† can be attained. The other system parameters such as repair man-hours are transformed into weight as detailed in Chapter IV. The research results are in the form of a mathematical model which enables the calculation of the number of each specific module to carry on a mission as spares, from a given set of candidate modules, for an optimum design.

An obviously important aspect of achieving desired space mission reliability by component replacement is the degree to which common components can be used in the spacecraft subsystems. Greater commonality among replacement

---

†The reliability, R(t), of a system is the probability that the system is operating at time t.

modules leads to lower spares requirements for achieving the desired reliability. Some effects of increased commonality are indicated in Chapter IV, and provision is made for consideration of commonality at all levels in the solution procedure presented in Chapter V.

## Statement of the Problem

A basic assumption of this discussion is that the spacecraft was first designed as being divided into the lowest level of component parts which it is feasible to replace. In order to replace parts at this lowest feasible (smallest part) level, each part or module† must be designed with the proper packaging, connectors, and accessibility in order to perform the replacement in a reasonable amount of time and with relatively simple replacement procedures. Equipment must be provided to determine which module is to be replaced (fault isolation) when a malfunction has occurred. Replacement effort in man-hours provided by the crew must also be considered.

By aggregation of the lowest level of replacement modules, thus constructing larger modular units, a reduction in the weight of the basic components can

---

†The term module will be used to describe any aggregation of parts or components considered for replacement. Hence, even the lowest level of component parts which it is feasible to replace may be called a "module".

obviously be accomplished. This will result in reduced fault isolation equipment, connectors, and accessibility problems (less volume and weight required). A module composed of an aggregation of several smaller components will also, in general, have a higher reliability than the configuration represented by having the smaller components as the replaceable units of the system. However, the increased module size and corresponding decreased commonality of modules can result in a larger weight of spares to achieve the desired reliability.

The problem is to determine the candidate modules to be designed for replacement and the number of spares of each module to be carried in order to minimize a specific spacecraft's weight and achieve a given mission reliability.

Mathematical statements of the above problem with various assumptions and considerations will be presented in Chapter IV.

## General Assumptions

In the foregoing discussion, a number of assumptions were implied and the details omitted. In this section a brief statement of the general assumptions are listed in order to complete the introductory statement of this research topic in a non-mathematical context. A summary

of the assumptions include:

1. A hierarchy of candidate module combinations (feasible packaging structure) can be developed. The number of module size levels to be considered is expected to be small (four or fewer).
2. The reliability of spares not in use is not degraded.
3. All component failures are statistically independent; this is assumed to imply functional independence.
4. The replacement action does not cause component failures.
5. The reliability of a module formed by an aggregation of components is greater than the reliability of the same set of components designed for individual replacement.
6. Each of the parameters affecting the desirability of selecting a specific replacement module for the system can be converted to a common parameter such as weight. Examples of such parameters are the weight of the test equipment, replacement time, volume required for access, and fault isolation time.
7. All of the modules in the single thread system are required for the successful completion of

the mission.

8. Neither repair nor replacement of components within a failed module is permissible.
9. The models developed will provide for random failures only, as opposed to scheduled maintenance.
10. The values of the parameters required by the models can be estimated.
11. Functional redundancy, the utilization of one spacecraft system to perform the function of another system in a backup mode, is not considered.

# CHAPTER II

## REVIEW OF THE LITERATURE

The literature search was conducted in three primary areas:

1. redundancy optimization problems related to this research,
2. optimum module size studies, and
3. branch and bound approaches to the solution of integer optimization problems.

Most of the literature review will be devoted to a review and analysis of work accomplished in redundancy optimization, since the work in this research is closely associated with that area and uses it as a basis for extending much of the reported work.

The theory proposed in the literature cited has not been operationally verified. Since the topic of this research is one which is under consideration primarily for future space missions, and since long duration space missions have not yet been accomplished, it should be realized that many of the basic questions concerning the hardware and mission requirements are as yet unanswered. Whenever possible, the guidelines and assumptions which correspond to the results reported will be made explicit in the literature review, since the validity of such

assumptions preclude the usefulness of the research conclusions.

## Redundancy Optimization Problems

A topic which is closely related to this research and one which is inherent in the solution to the problem posed herein is that of optimum redundancy. The problem of determining the optimum module size for extended space missions can be viewed as an optimum redundancy problem with the complications of common components, redundancies added in the form of spares, and a problem of selecting the system to be optimized from among candidate designs.

One of the early papers concerned with finding an analytical solution for the number of spare parts of each type which would maximize a measure of system performance subject to a fixed expenditure for spares was presented by Geisler and Karr [20] in 1956. The specific measure of system performance used as a criterion for selecting spare parts is the expected number of shortages, that is requirements for parts for which all of the spares have been consumed. Each shortage is also weighted by the estimated criticality of the part, which is the term given to the seriousness of a shortage of one part relative to the seriousness of shortages of other parts. The solution to the problem as stated is obtained by assuming that the

number of parts of each type to be spared is a continuous quantity. Therefore, it is possible to stock fractional units, and define the problem as follows:

$$\underset{S_i \geq 0}{\text{Minimize}} \sum_{i=1}^{m} C_i \int_{S_i}^{\infty} (x - S_i)\, p_i(x)\, dx,$$

$$\text{subject to} \sum_{i=1}^{m} \omega_i S_i = w,$$

where $m$ = the number of parts eligible to be spared,

$p_i(x)$ = the probability density function for a demand of $x$ units of the $i^{th}$ part during the time of concern,

$S_i$ = the quantity of the $i^{th}$ part to be spared,

$C_i$ = the criticality factor for the $i^{th}$ part,

$\omega_i$ = the unit weight (cost) of the $i^{th}$ part,

$w$ = the total allowable weight (cost) of the spares selected.

The problem is solved by using the classical Lagrange multiplier approach by solving for each value of $S_i$ by an iterative technique and rounding off to obtain integer values for $S_i$. A second solution method is to compute the ratio

$$\frac{C_i}{\omega_i} \sum_{x=a}^{\infty} p_i(x)$$

for successive values of a = 1, 2, . . ., for each of the m parts. The sum of this series of decreasing ratios for each part will give the probability that the $i^{th}$ unit of that part will be required. By ordering the ratios for all parts by decreasing value, the parts are selected from the top of the list until the weight (cost) constraint is reached. Both of these solution procedures result in an approximation to the optimum of the problem as stated.

Proschan and Black [52] in 1959 solved an optimum redundancy (or spares) problem similar to that of Geisler and Karr. Explicit formulas for computing an approximate solution in the case of exponential failure distributions were developed. The second solution technique of Geisler and Karr and the method of Proschan and Black have since been referred to as the delta or one-at-a-time method. While these approaches do not guarantee a precise optimum, Barlow and Proschan [3], page 585, state that the loss in protection due to the approximate character of the solution procedure will generally be small, especially when there are a large number of component types involved in the computation.

The essence of the delta method is to select the spares or redundant components such that each successive selection maximizes the incremental increase in system reliability per unit of resource expended for that

component. Stating the above procedure mathematically, the system reliability is first written as [52]:

$$P(n) = \prod_{i=1}^{k} P_i(n_i),$$

where $P_i(n_i)$ = the probability that $n_i$ spares of type i will be adequate, $n = (n_1, \ldots, n_k)$.

For the case of an exponential failure distribution, the reliability of a given type of component with redundant units is:

$$P_i(n_i) = \sum_{x=0}^{n_i} \frac{(\lambda_i)^x}{x!} e^{-\lambda_i},$$

where $\lambda_i$ is the parameter of the Poisson distribution which appropriately represents the sum of the failure rates of the components of type i multiplied by the time for which the system is to operate. By defining:

$$R_i(n_i) = \ln P_i(n_i), \text{ and } R(n) = \ln P(n),$$

the maximization of P(n) is equivalent to maximizing R(n). The increase in system reliability,

$$\Delta R_i(m) = R_i(m + 1) - R_i(m),$$

due to the addition of the $(m + 1)$st spare of type i can

be computed easily. The ratio which yields the greatest increase in system reliability per unit of resource,

$$\frac{\Delta R_i(n_i)}{c_i},$$

is calculated, where $c_i$ is the weight (cost) of the $i^{th}$ component. The components are then added one at a time, calculating a new value of the above ratio for each component after each addition of a component, and adding the successive component for which the ratio is a maximum. The process is continued until the allowable quantity of resource is utilized.

An exact solution to the optimum redundancy problem was reported in an early paper by Bellman and Dreyfus [6], 1959, in which the reliability is maximized subject to constraints of cost and weight.

Another dynamic programming approach was taken by Kettelle [34] to obtain a solution to the basic problem of the least-cost allocation of redundancy while constraining the system reliability. Proschan and Bray [53] extended the work of Kettelle in a very similar dynamic programming formulation of the optimal redundancy problem with multiple linear constraints. In a recent work [10], Carey elaborates on the dynamic programming approach taken by Bellman and Dreyfus for the two

constraint case.

As an illustration of a dynamic programming formulation of an optimum redundancy problem, a particularization is presented here.

Consider a system of N components in series which is to have $m_i$ components placed in parallel with the $i^{th}$ component, i = 1, . . ., N. An illustration of such a system is shown in Figure 2.1.

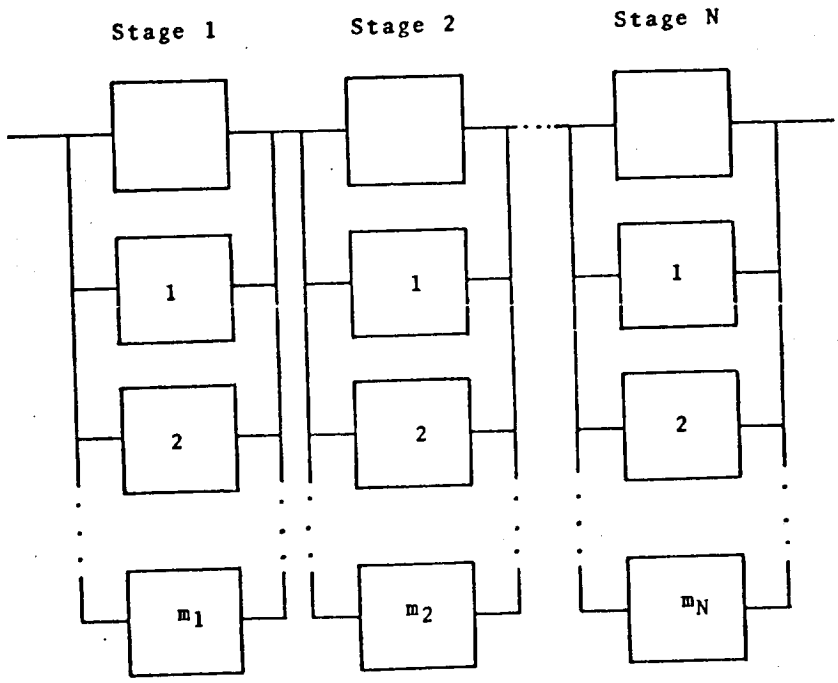


Figure 2.1.--Illustration of a series-parallel system.

The reliability for each stage is represented by $\phi_i(m_i)$. The system reliability, to be maximized, is the product of the stage reliabilities:

$$R_s = \prod_{i=1}^{N} \phi_i(m_i).$$

A single cost constraint can be stated as:

$$\sum_{i=1}^{N} c_i \, m_i \leq C,$$

where $c_i$ is the cost of component i and C is the total allowable cost for the system. The dynamic programming recursive relationship is:

$$f_n(c) = \max_{0 \leq m_n \leq [c/c_n]} \{\phi_n(m_n) \; f_{n-1} \; (C - c_n m_n)\}, \; n=2, \; 3, \; \ldots, \; N,$$

and $$f_1(c) = \phi_1([c/c_1]),$$

where the bracketed quantity $[c/c_n]$ means the largest integer less than or equal to $c/c_n$.

Another formulation of the optimum redundancy problem was recently reported by Tillman and Liittschwager [59], which extends the problem to consideration of non-linear separable constraints. This extension is developed by linearizing the constraints and the objective function, and using integer programming on the transformed problem.

The integer programming approach also allows consideration of a limited case of alternative designs.

The problem of optimum sparing has also been considered from the standpoint of maintenance float requirements [64]. Maintenance float is defined to be the spares required for a system to perform satisfactorily for an unlimited time with repair and replacement of spares in the float inventory. A simplified binomial form of the statement of system availability is used as the objective function, and the optimum maintenance float requirements for a system with common components is found using dynamic programming.

There have been two recent reports on solutions to a version of the optimum redundancy problem [17,44]. These two approaches have the common characteristic of assuming the number of redundancies is a continuous variable, solving for the non-integer optimum, and rounding off the solution thus obtained to the nearest integer value. It is easily seen that these types of approaches can result in a solution which is quite poor relative to the true optimum, especially where the number of redundant components of each type at optimum is small, such as 0, 1, or 2.

There have been several modifications of the optimum redundancy problems described above. Because of the wide-

spread interest in such problems, reference is sometimes made to the "classical optimum redundancy problem", which usually refers to the problem cited in [6]. One additional modification to the optimum redundancy problem includes consideration of the effect of inadvertent operation of a component in parallel or standby, which then results in the optimum unconstrained redundancy problem having a finite solution, namely the number of redundancies of each type for which the reliability is a maximum.

The following chapter will discuss the application of some of the above redundancy optimization problems to manned spacecraft systems.

## Application of Redundancy Optimization to Space Systems

One of the most extensive studies directed toward the optimization of spares requirements and other maintainability considerations was performed by the Space Division of the Boeing Company under a research contract with the National Aeronautics and Space Administration [42]. Since this research deals with many of the same aspects of long duration space flight as the Boeing study, an extensive review will be made of selected portions of the study.

A basic question investigated by the Boeing study is to determine the optimum spares and redundancy require-

ments for specific long duration manned space missions with a specific spacecraft design. A computer program given the acronym MARCEP for Maintainability and Reliability Cost Effectiveness Program [41], developed by the Boeing Company, was used to determine the redundancy allocations.

The program is entered with a single thread system, which contains the minimum number of components required to perform the system function, and various parameters associated with each component. Components are added by the delta method (discussed in a previous section), and may be placed in either standby or parallel configuration, or may be carried as spares. The decision between parallel and standby redundancy is made on the basis of a selection criterion, with an assumed switch reliability included in the decision. A test on the input data associated with each component is made to determine whether the expected replacement time, allowable downtime, and accessibility characteristics are such that redundant components can be added in the form of spares. If, for any reason, spares cannot be used as the only means of redundancy, parallel or standby redundancy must be utilized first, and then spares may be added if additional reliability is required.

In order to calculate the reliability of the system, and thus the increase in reliability due to the addition

of a redundant component, the reliability of a single component with n parallel components (including the single thread unit) and m spares was developed.† A pictorial representation of such a configuration, called a "cell", is shown in Figure 2.2 below.

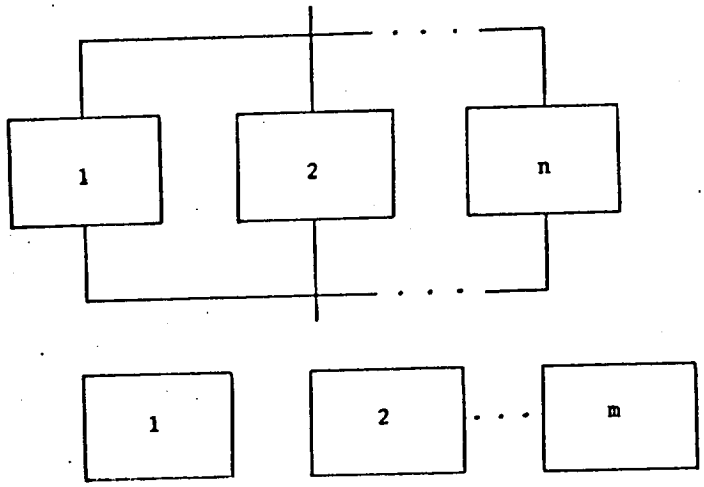


Figure 2.2.--Cell configuration.

By using the assumption that the time between failures is distributed according to the negative exponential distribution, the following equation was developed by the Boeing Company for the reliability of the above cell.

---

†In order to simplify the presentation, further discussion will exclude standby redundancy. The method of handling standby components is to treat them as spares with the switch reliability modifying the reliability of each.

$$R_{cell} = \sum_{i=0}^{m} \frac{(n\lambda t)^i}{i!} e^{-n\lambda t} +$$

$$\sum_{j=1}^{n-1} (-1)^{j+1} \binom{n}{j} \left[\frac{n}{n-j}\right]^m e^{-j\lambda t} \left[\sum_{k=m+1}^{\infty} \frac{[(n-j)\lambda t]^k e^{-(n-j)\lambda t}}{k!}\right], \quad (2.1)$$

where $\lambda$ is the failure rate of the single component, and t is mission time.

An equivalent expression to the preceeding was derived by Wilken and Langford [66] using a quite different approach from that reported by Boeing.

The above expression does not provide for calculating the reliability of a system which has more than one, N, of the same component in the single thread system (commonality case). In such a case, it is assumed that the spares can be used to replace any one of the n x N parallel components and that replacement occurs immediately upon failure of a component. This configuration is represented in Figure 2.3.

The development of the method programmed in MARCEP to calculate a value for the reliability of the configuration represented by Figure 2.3 is as follows. It is first noted that if the cellular structure is maintained by providing for each spare to be used only within a specified cell, a

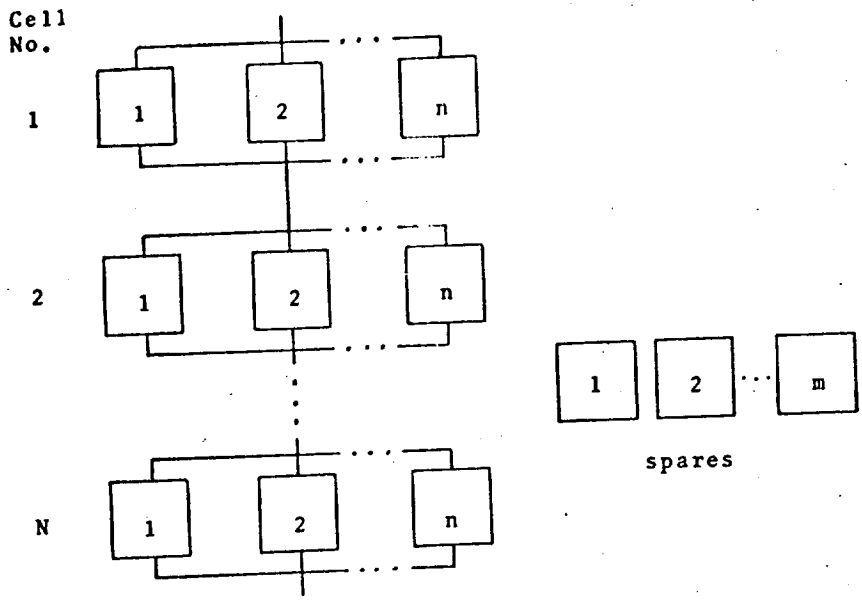


Figure 2.3.--Commonality configuration.

gross underestimate of the true reliability of the configuration will result. The development proceeds by attempting to identify the possible ways in which the spares can be utilized. For example, if there are three redundancy cells (N=3), and two spares (m=2), there are two possible uses of the spares: either two spares on one cell and zero spares on the other two cells, or one spare on each of two cells and zero spares on the third cell. The reliability of each of the configurations which would result if each spare were limited in use to a

single cellular structure is then computed (using equation 2.1) for each of the possible uses previously determined. Figure 2.4 illustrates the six configurations possible for three cells and two spares.

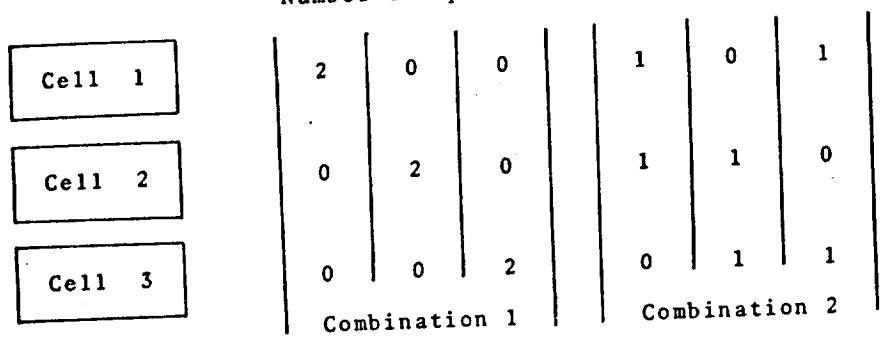


Figure 2.4.--Possible consumption of two spares with three cells.

The next step is to weight the reliabilities computed for each of the combinations shown in Figure 2.4 according to the frequency of occurrence of each possible consumption of the spares. The sum of the weights is equal to 1.0. The reliability of the system of Figure 2.3 is then computed as the weighted sum of the reliabilities computed for each combination of spares utilization.

The calculation of the values used to weight the reliabilities computed for each combination of spares utilization is best illustrated by reference to the case

of three cells and two spares illustrated in Figure 2.4. The reliability of the configuration which would result if only Combination 1 were allowed, $R_1$, is first computed. Define $r_{ij}$ as the reliability computed from Equation 2.1 by assignment of i spares to j of the N cells. The reliability for Combination 1 is:

$$R_1 = (r_{21})\ (r_{02}).$$

The reliability, $R_2$, computed for the case of allowing only Combination 2 is:

$$R_2 = (r_{12})\ (r_{01}).$$

The reliability of the entire configuration shown in Figure 2.4 for N=3 and m=2 is then computed as:

$$R = \frac{A\,R_1 + B\,R_2}{A + B},$$

where A and B are the frequencies of occurrence of combinations 1 and 2 respectively.

The equation for calculating the frequencies of occurrence for the general case of N cells and m spares is not explicitly derived. The weights used for any N and for m=2 are given as $\frac{1}{N}$ and $(1 - \frac{1}{N})$. The values for A and B for the calculation of the corresponding system reliability are A = N and B = N(N-1). Hence, the

general expression given for the system reliability with N cells, $N \geq 2$, and two spares is:

$$R = \frac{1}{N} (r_{0,N-1} \cdot r_{21}) + (1 - \frac{1}{N}) (r_{0,N-2} \cdot r_{12}).$$

In all of the foregoing discussion, the purpose has been to merely explain the methods used by the MARCEP program as explained in [41]. A brief critique of this method of calculating the reliability for the commonality case shown in Figure 2.3 is inserted since, in the course of this research, the above development was found to be incorrect. A derivation of the reliability of the system of Figure 2.3 for cells having two components in parallel and for any value of m and N is presented in Chapter III.

First, it is noted that the weights or frequencies of occurrence of each possible use of spares should be calculated by considering the number of cells having 0, 1, 2, . . ., m spares assigned to them as a permutation of indistinguishable objects. This approach leads to the frequency of occurrence of each possible use of two spares to be:

1. 2 spares to 1 cell and 0 spares to N-1 cells,

$$\text{frequency of occurrence} = \frac{N!}{1!\ (N - 1)!}.$$

2. 1 spare to 2 cells and 0 spares to N-2 cells,

$$\text{frequency of occurrence} = \frac{N!}{2!\,(N-2)!}\,.$$

For the above case of N=3 and m=2,

$$A = \frac{3!}{1!\,(2)!} = 3, \quad B = \frac{3!}{2!\,(1)!} = 3,$$

which assigns a relative weighting of 1/2 for both uses of the two spares instead of the ratios 1/3 and 2/3 which are calculated for the two cases $r_{02}r_{21}$ and $r_{01}r_{12}$, respectively, by the MARCEP program.

The second, and more important point to be made concerning the reliability calculations for the commonality case as made by the MARCEP program is that no weighting of the reliabilities calculated by considering the spares as assigned to each cell according to their possible use can result in the correct reliability of the system, as long as the sum of the weights is unity. This assertion is easily shown by considering a weighting such that the use of spares which results in the highest reliability of the system is given a weight of one, and all other possibilities a weight of zero. Even in this case which maximizes the calculated reliability, R, it is obvious that the procedure would result in a lower reliability than the system actually has, since each spare is restricted as to

its application within the cellular system. The MARCEP procedure results in a still lower reliability because a finite weighting is given to each of the cells including those given a weight of zero in the above procedure.

After the spares required for the particular mission have been determined using the MARCEP program, the resulting configuration was used as an input to a simulation model. By simulating a mission having random failures, the effects of maintenance times, spares weights, mission duration, system reliability and other parameters were determined. Outputs of the simulation include repair man-hours, number of crewmen involved at any one time, times to repair, and related information.

Some of the conclusions and recommendations resulting from the Boeing study which are especially pertinent to this research are the following.

One of the general results of the study was the conclusion that a single maintenance philosophy could not be effectively applied to all of the spacecraft subsystems. It was determined that the maintainability characteristics must be considered as part of the design criteria and that they should be applied at the component level.

More specific conclusions include the determination that unscheduled maintenance requirements constitute a relatively low demand upon the total crew workload. A

simulation of the repair activities indicated, however, that increasing the mean-times-to-repair can result in significant increases in the weight added for redundant components and spares.

It was determined that commonality of components should be one of the most important design objectives. Increased commonality was found to greatly reduce spares requirements, program costs, and training. Modularization of components was also recommended as an effective way to reduce spares requirements and maintenance workload.

The general maintenance philosophy of removing and replacing failed components was recommended for subsystem design, as opposed to repair of parts or automatic switching as a form of redundancy. An exception to this concept, which minimizes spares, overall weight, and cost, is noted for the case of only one spare required, in which case it is recommended that the standby redundancy configuration be utilized.

A premise considered to be validated by this study is that maintenance will be necessary for long duration manned space missions, and also that it will be desirable and practical. The desirability of maintenance is attributed to the weight reduction which results from replacement as opposed to other types of redundancy.

Another study in which sparing and maintenance requirements for long space missions were considered was performed by the Space and Information Systems Division of North American Aviation for the National Aeronautics and Space Administration [43]. One of the characteristics of this study is that the reliability requirements for a specific mission were apportioned by subsystems such as environmental control and electrical power. These were further subdivided to major units within several of the subsystems, such as the computer in the guidance and navigation subsystem. The actual selection of spares and other redundancies in this study were accomplished by heuristics with detailed analysis of subsystem functions to determine the criticality and accessibility of various components. The primary concern was to achieve a desired subsystem reliability in a practical way with little regard for mathematical optimization.

In a study directed toward an analysis of the space-craft systems in the Apollo Program [65], a selection procedure is described which uses the delta method as its basic technique. The selection index includes not only the basic reliability and weight parameters, but a subjectively determined criticality index. Three selection criteria are suggested, each one being a variation of a value calculated by combining the system parameters. Briefly, the three

criteria are:

1. The expected difference between the spares required and the spares available.
2. The expected number of module types for which there will not be a sufficient number of spares available.
3. The probability that the spares aboard the spacecraft will be sufficient to cover all failures which occur during a mission.

As in all of the problems described in this section, the procedure is developed only to apply to a specific design, rather than using the reliability requirements to determine the optimum design.

In a document directed toward determining the maintenance activities which may be required during an extended space mission, Black [8] considers such factors as weight, volume, fault isolation, repairability, accessibility, and crew training. Optimization of redundancy with regard to type of redundancy is deemed to be desirable, but specific procedures for obtaining an optimum design are not presented.

Another conceptual approach to the problem of attaining required reliability for long space missions is presented in [11]. This report considers the possibility of repair of malfunctioned components as well as the

various types of redundancy. As in [8], however, procedures are not proposed by which the required trade-offs between repair and redundancy can be accomplished. The primary contribution of this report is to define in a general way the reliability problems associated with long space missions and to propose that certain studies are required to analyze the design alternatives.

In a similar manner to the report by Carpenter [11], Peterson [50] discusses an overall philosophy and management approach to the consideration of maintenance in future space systems. A maintainability objective is also proposed but quantitative analyses for optimization are not presented.

Another study [31] was accomplished to develop analysis procedures for determining maintenance requirements for long space missions. An extensive computer program was developed for this purpose; but, as in the two previous cases [11,50], the procedures described are in the form of a logical plan for in-space maintenance as opposed to a quantitative optimization approach.

## Optimum Module Size

The literature search revealed that the specific optimum module size problem posed for the central problem of this research has not been reported on to date. The

references cited in this section are those which refer to optimum module size either in the title or by describing a problem concerned with levels of redundancy in any manner.

Goldman and Slattery [26] devoted a chapter of their book to a discussion of optimum module size with respect to minimizing the cost of a maintained system. The implications of modularization are discussed. The primary advantage of modularization is reduced downtime. Also, with increased miniaturization of electronic parts, the future trend will be toward design of the replacement modules to be discarded, thus eliminating repair in the maintenance cycle. However, it is noted that if present day equipment is analyzed for repair versus discard decisions, repair will often appear to be better simply because the module complexity is too high; that is, each module has too many parts in it.

The discussion [26] of optimum module size is directed toward electronic systems in which the size of a module is measured simply by the number of parts in it. An optimum module size is defined as ([26] page 129), "... that which minimizes, for desired performance capability, the expected annual costs of purchasing and operating the equipment over its program life." It is further stated that the optimum module size concept is dependent upon "... an idealized situation in which the

equipment is homogeneous throughout with regard to the kind of components employed", ([26], page 129). Accordingly, the models and solutions presented are based on the concept of a system of homogeneous parts which can be aggregated in any manner desired to construct modules. Some of the costs which are minimized in determining the optimum module size are, for example, manpower costs, inventory costs, and initial purchase cost for long term operation of the system.

From the above discussion, it is apparent that the optimum module size problem as posed in [26] bears little resemblance to the problem of this research. Descriptions of the current literature on optimum module size is included here for completeness, even though the actual problem approach taken herein is more closely associated with the literature under the optimum redundancy problem title.

Flehinger [19] has reported on an investigation to determine the optimum level at which redundancy should be applied. The application is to computers in which the levels of redundancy refer to the number of identical units to aggregate in order to maximize reliability. The analysis is concerned only with standby redundancy in which a component is switched into use to replace a failed component. A failure detector associated with each unit

indicates the failure of a unit. The analysis is performed on a time independent basis. The effect of imperfect switching is also considered, which leads to an unconstrained maximum attainable reliability as a function of the number of redundant units. It is concluded that for initially reliable units the level at which redundancy is applied is of less importance than for initially unreliable units. The decision concerning the level for application of redundancy is left to judgement, weighing such factors as cost, convenience, and the reliability of available switches and error detectors.

A consideration of the aggregation of components into modules is considered on a module-by-module basis, independent of the rest of the system by Joeseph [33]. This report is concerned with the best way to package components in a spacecraft with the primary criterion being the most reliable way. Various types of packaging structures are considered, and discussed as they relate to the reliability of the module. As an example of the generality of the study results, the following conclusion concerning the packaging of digital equipment is cited (page 425): "The optimum size of encapsulated subassembly was found to be approximately a 5x7 inch package consisting of one level of cordwood modules and/or integrated circuit stacks." It is again obvious that the meaning of "optimum module size" is

quite different in [33] then in the context of this research.

## Branch and Bound Solution Methods

The formulation of the optimum module size problem about which this research is centered is formulated in Chapter IV as an integer programming problem. Since the algorithm developed to solve this integer programming formulation can be viewed as being of the type characterized as "branch and bound", the literature concerning such solution techniques is summarized here.

In recent years, there has been considerable attention given to the solution of integer programming problems by a solution approach called "branch and bound". This approach, which Geoffrion [22] calls "implicit enumeration", was reported by Balas [1] and Glover [23] in 1965. In a rather comprehensive survey of integer programming solution procedures, Balinsky [2] in 1965, indicated the potential of such methods when he wrote: "... various clever methods of enumeration can be the most efficacious means existent by which to obtain solutions to practical problems".

As the name implies, branch and bound methods involve consideration of groups of solutions from a tree of logical possibilities, determining those solution groups

which are dominated, ruling them out of consideration, and subdividing those groups which are not dominated. The tests used to rule out groups previously created are then repeated on the new groups (branches) until all solutions have been (implicitly) considered.

In order to discuss some of the characteristics of branch and bound algorithms further, it will be convenient to introduce some terminology and definitions. It will also be helpful to state the integer programming problem as having, if the integer requirements were removed, a linear objective and linear constraints. Most of the current algorithms are directed toward these problems. It will become apparent as the discussion proceeds that the branch and bound approach in general is not confined to consideration of problems with only linear functions.

An additional particularization of the general integer programming problem which will be used as the basis for this discussion is that all of the variables are restricted to be integers. As with the linearity assumption, the branch and bound solution method does not require all variables to be integers, but the solution algorithm developed in Chapter IV is of this type, and discussions will be simplified by this specification.

The linear all-integer programming problem can be stated as follows:

$$\text{Minimize} \quad Z = c\,X$$
$$\text{subject to} \quad A\,X \geq b \tag{2.2}$$
$$x_j \text{ non-negative integer, } j = 1, 2, \ldots, n.$$

Some of the branch and bound algorithms assume that all of the integer variables are restricted to the values 0 and 1 [1,22,23]. There is no loss of generality from allowing only this binary set of values if the $x_j$ are bounded, since the transformation

$$x_j = \sum_{i=0}^{k_j} 2^i y_{ij}, \qquad y_{ij} = 0 \text{ or } 1,$$

is possible, where $k_j$ is sufficiently large to allow representation of all feasible values of $x_j$. An alternative transformation which may be used is

$$x_j = \sum_{i=0}^{n} y_{ij}, \qquad y_{ij} = 0 \text{ or } 1,$$

which results in a larger number of binary variables, but utilizes a somewhat simpler transformation. The properties of the latter transformation may be advantageous in the problem formulation (for example, see [60]).

To facilitate further discussion, a list of definitions used by Geoffrion [22] is given below. It should be noted that other authors use different terms having the

same meaning as the following, but for continuity of presentation the following terms will be used in the discussion of various approaches when possible.

1. Partial solution: Assignment of an integer value to a subset of the n variables.
2. Free variables: Those variables to which a value is not assigned.
3. Completion of S: A solution determined by S plus a specification of values to the free variables.
4. Fathom S: Either (1) find the best feasible completion of S, or (2) discover that no solution is both feasible and better than the incumbent (a feasible solution previously found).

The basic procedure.

A very general sequence of steps is given below to describe the underlying concepts of the branch and bound approach.

1. Start with the initial partial solution, $S^0$, being an empty set. That is, all of the n variables are free.
2. If all partial solutions which can be generated by the variables represented in S are fathomed, the solution has been found. Otherwise,
3. Augment S by assigning a value to one of the free

variables. The value assigned is such that a previously fathomed partial solution, S, is not repeated and the value assigned is not an element of a solution previously ruled out by fathoming S.

4. Attempt to fathom the partial solution just augmented. Return to step 2.

It should be noted that since there are a finite number of solutions to the bounded integer programming problem, and since step 3 provides for non-repetition of fathomed partial solutions, the above general procedure is convergent. Most branch and bound algorithms are special cases of the above general procedure. There are various procedures for augmenting S as outlined in step 3. If a tree of solutions is visualized, this step can also be referred to as branching. An example of such a tree for a problem having three variables, $x_1$, $x_2$, $x_3$, is illustrated in Figure 2.5. The tree is constructed to allow the solutions $x_1 = 0, 1$; $x_2 = 1, 2, 3$; $x_3 = 0, 1$. Each node in the tree represents a partial solution, S, and the terminal nodes each represent a complete solution, with their associable unique path from the origin determining the values of each variable for that solution.

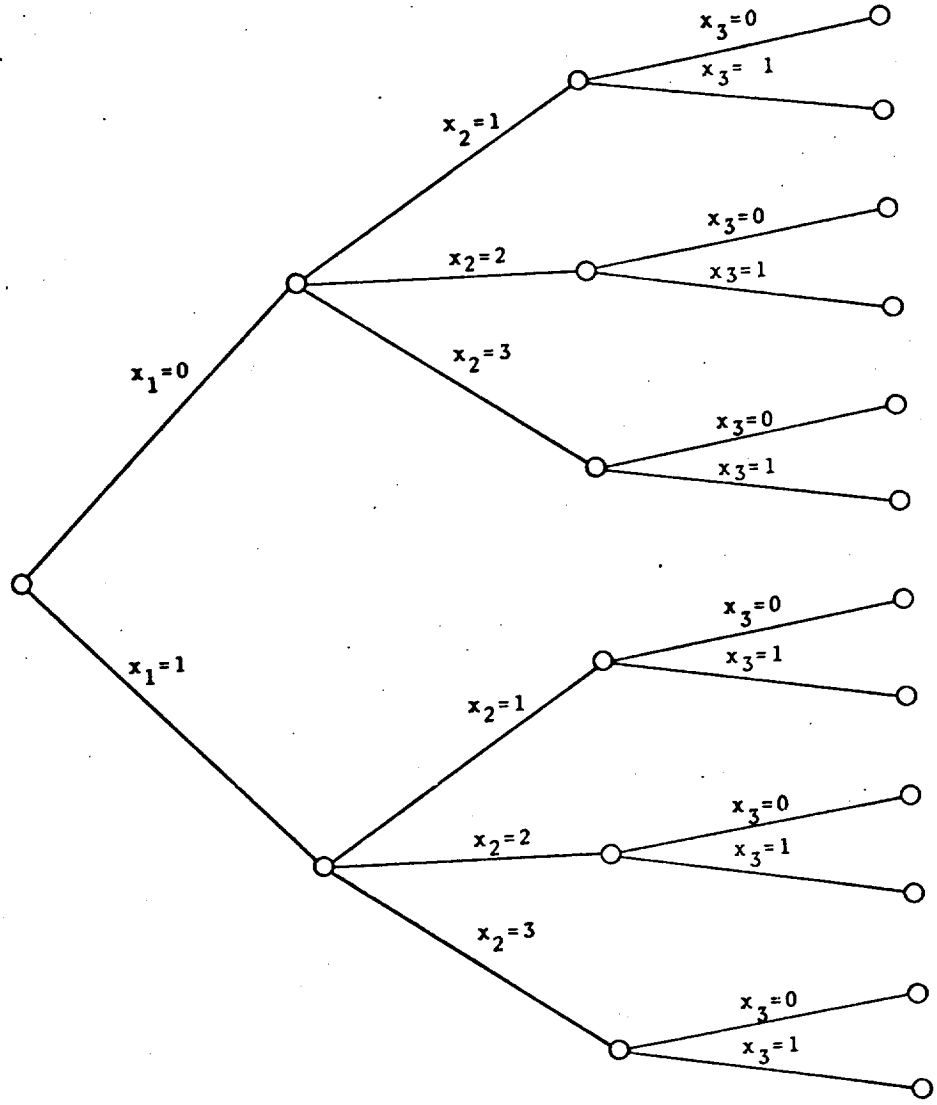


Figure 2.5.--Example of a solution tree.

Two classifications of branching procedures generalized by Lawler and Wood [38] are discussed below in general terms.

Methods of branching.

The simpler of the two branching procedures is to branch from the latest partial solution generated. The basic idea of this procedure, referring to the solution tree shown in Figure 2.5, is to generate the succeeding nodes of a given node until a partial solution is fathomed, before generating a node that is not a successor to the most recent one generated. In this way, new problems are "stacked", and at each addition of a new problem to the stack, an attempt is made to fathom the corresponding partial solution. If it cannot be fathomed, branching is performed on it. If the problem on top of the stack can be fathomed, it, along with all completions of the corresponding partial solution, is dropped from the record, and the next problem on the top of the stack is examined.

An important advantage to the above procedure is that the memory requirements are kept to a minimum, since only a small portion of the solution tree must be kept in memory at any one time. The maximum memory required depends upon the length of the longest path from the origin to a terminal node of the solution tree.

The other general approach to branching is termed

branching from the lowest bound. This procedure specifies that the next node added to the solution tree is added to the partial solution which has the lowest bound. If the part of the solution tree which must be generated to solve the problem is unique, then the branches created by this method would also be required under any other branching method. The primary disadvantage of this method is the large memory requirement in order to maintain all the sets of candidate partial solutions from which the one having the lowest bound is chosen. This lowest bound is used as a basis to perform the branching operation.

Combinations of the above methods may also be used. For example, one may branch from the newest active bounding problem until a partial solution is fathomed and then, instead of branching from the fathomed partial solution, branching may be carried out on the partial solution with the lowest bound. From this lowest bound partial solution, branching may then revert to the policy of branching from the newest active bounding problem, until another partial solution is fathomed.

Trial solutions.

There are three common types of trial solutions which are actually completions of S. The possibilities for other types of trial solutions are many, and the particular one(s) selected for a specific algorithm depend upon the

character of the problem and the strategy employed in the branch and bound procedure. The purpose of trial solutions is to attempt to fathom a partial solution or to bound a subset of the solutions in order to avoid further consideration of the subset. Three of the most common types of trial solutions are:

1. Set all of the free variables to 0.
2. Find the completion of S that is the best feasible completion to problem (2.2) or to a problem resulting from relaxing some of the constraints of problem (2.2).
3. Use the completion as given in (2) except any non-integer components of the completion of S are rounded off to integer values.

If a trial solution of type 2 is found, it is often possible to use it directly to fathom S by having found the best feasible solution or by having shown that no feasible solution is better than the incumbent. A specification of this type of trial solution will be used in the algorithm developed in Chapter V. Trial solutions of type 2 are also used to carry out the method of branching from the lowest bound previously described. With regard to the second part of trial solution type 2, the relaxation of constraints is often accomplished by removing integer restrictions [21] or by simply not

considering some or all of the constraints at a certain point in the solution procedure [1]. In any event, relaxing constraints does not exclude any solutions which are feasible for the original problem, but in general may enlarge the solution tree which must be searched.

## Summary

The solution techniques applied to redundancy optimization problems can be divided into three categories:

1. those which use heuristics to arrive at an allocation of redundant components which is 'good' by some predetermined logic;
2. those which use a mathematical expression for the selection criterion, but which approximate the optimum for the stated objective;
3. those which utilize a model such that a feasible solution is found which optimizes the objective function.

The heuristic techniques have an advantage in being able to consider various subjective complications such as ranked criticality of the components, complexity of design and difficulty of accessibility. Since there is a lack of rigor in such methods, however, there is a lack of incentive to strive for the best possible estimation of parameters and of a measure of goodness of the resultant

system.

The methods of redundancy optimization which rely on approximation techniques are generally the easiest to program and compute. However, in the case of spacecraft design alternatives, the need exists to utilize design decisions to the fullest extent possible.

In addition to resulting in a precise solution, the method of optimization of a mathematical model requires the particular problem to be rigorously structured. The analysis involved in developing the model can lead to insight into the character of the system to be optimized and thus provide the basis for consideration of other design alternatives and for generalizing design concepts. For the purpose of exploring the various design alternatives for long mission spacecraft, the problem insight provided by the rigorously structured model may justify the effort involved even if the optimum solution to the model cannot be found.

A brief statement of the literature concerned with the optimum module size problem is that the problems posed thus far have only considered the case of aggregation of homogeneous units of equipment. The types of problems thus approached have little connection with the problem posed in this research.

The solution technique of branch and bound seems to

be most promising for solving problems having discrete variables such as in redundancy optimization problems. Of particular interest are the cases where the initial problem variables are binary in nature, as are some of the variables described in Chapter IV for the optimum module size problem. Also of interest are those in which the reliability statement for the system being analyzed is very complex. The branch and bound methods are particularly suited to solving both of these cases.

# CHAPTER III

## DEVELOPMENT OF THE RELIABILITY OF REDUNDANT SYSTEMS

In order to develop any of the models considered for determining the optimum module size, it is necessary to state mathematically the reliability of the system. In this chapter the reliability of various types of systems which may be used as a basis for considering the optimum module size problem will be developed. As a result of extensive work with spared systems having common components in parallel, some comments concerning the method of derivation of the reliability of complex systems are also presented. Illustrations are given to demonstrate the effect of having common components in spared systems and of components in parallel in a system with spares. Generalizations concerning the design of spared systems are presented and illustrated by curves showing the effect of two types of redundancy on system reliability.

In Chapter IV, certain of the statements of reliability developed in this chapter will be used as constraints in the formulation of the optimum module size problem.

## Time Independent

A simplifying assumption which facilitates initial discussions is that component failures are independent of time. Another interpretation leading to the same reliability statement is that components are equally likely to fail in use and as spares in the passive condition before being used. With the latter assumption, any failure distribution will lead to the same reliability function, for specified individual component reliabilities.

Suppose the basic system for which the time independent reliability is to be derived consists of $N_i$ components of type i, i = 1, 2, ..., K, all of which must operate for the system to survive. Let the basic system be augmented with $m_i$ spares of type i, and denote the set of components of type i and the corresponding spares as the $i^{th}$ component set. The reliability, $P_i$, of the $i^{th}$ component set is the probability that (0, 1, ..., $m_i$) failures of components of type i occur, since the occurrence of more than $m_i$ failures of type i results in system failure. It is easily seen that the number of failures of component type i follows a binomial distribution, therefore:

$$P_i = \sum_{x=0}^{m_i} \binom{N_i+m_i}{x} p_i^x \, q_i^{N_i+m_i-x} , \qquad (3.1)$$

where $p_i$ = the probability of failure of component i, $q_i = 1 - p_i$. Since each component set must survive for the system to survive, the reliability of the system is:

$$R = \prod_{i=1}^{k} p_i. \qquad (3.2)$$

It should be noted that the quantities $p_i$ may be computed from time dependent distributions as the probability that the $i^{th}$ component fails before time t, where t is the required survival time.

## Exponential Failure Law With Spares

It is commonly assumed that the time between failures is distributed according to the exponential distribution. This assumption is usually made to render a problem mathematically tractable due to the resultant properties of the system. There is some basis for this assumption in electronic and other space systems in which failures are not due to wear, but are proportional to the time the system is exposed to an operational environment. An attempt will not be made to justify the assumption of an exponential failure distribution except that, under any other distribution, the problem would be intractable.

When the time to failure follows an exponential

distribution, it is a well known result that the failures occur according to the Poisson distribution (for example see [46], page 13). Stated mathematically, if f(t), the distribution of time to failure of a component is:

$$f(t) = \lambda e^{-\lambda t}, \tag{3.3}$$

where $\lambda$ is the number of failures per unit time. Then, assuming the component is renewed each time a failure occurs, the probability of n failures in time t is ([3], page 176):

$$Pr(x=n) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}. \tag{3.4}$$

If there are N components having failure rates $\lambda_1, \lambda_2, \ldots \lambda_n$, then ([3], page 176) the probability for n failures in time t is:

$$Pr(x=n) = \frac{(\Lambda t)^n}{n!} e^{-\Lambda t}, \tag{3.5}$$

where

$$\Lambda = \sum_{j=1}^{N} \lambda_j. \tag{3.6}$$

Assuming each set of like components of the $i^{th}$ type is augmented by $m_i$ spares and using the reasoning in the previous section, the system reliability is:

$$R(t) = \prod_{i=1}^{K} \left[ \sum_{x=0}^{m_i} \frac{(\Lambda_i t)^x}{x!} e^{-\Lambda_i t} \right], \qquad (3.7)$$

where K denotes the number of component sets in the system and t is the time for which the system is to operate. Note that if $\lambda_1 = \lambda_2 = \ldots = \lambda_N = \lambda$, equation (3.5) becomes:

$$Pr(x=n) = \frac{(N \lambda t)}{x!} e^{-N \lambda t}. \qquad (3.8)$$

## Parallel Components With Spares

In the two previous sections the reliabilities of single thread or basic systems backed up by spares were expressed. When the basic system contains components in parallel redundancy and is augmented by spares, the development of the reliability expression for the system is quite involved.

In Chapter II, equation (2.1), an expression for the reliability of a component set was presented. The component set, illustrated in Figure 2.2, consists of n like components in parallel, augmented by m spares. The discussion explained the incorrectness of the derivation in the referenced publication [41] when equation (2.1) was used to calculate the reliability of a system consisting of N cells of parallel components as shown in Figure 2.3.

Note that the spares can be used in any of the (n x N) positions to replace failed components.

An expression for the reliability of the system shown in Figure 2.3 for two components in parallel (n = 2) and for any value of m and N follows. The exponential distribution and instantaneous replacement of failed components with spares (when available) are assumptions.

The method of deriving the reliability expression is to write the transition matrix ([56], page 73), followed by the resulting differential equations for small values of m and N. The Laplace transform is used in solving the differential equations, and it is in the transformed state(s domain) that the generalization for any m and N is made. The procedure is demonstrated for N = 2, m = 3, as illustrated in Figure 3.1.

The notation ijk will denote that i components of cell 1 are in the failed state, j spares have failed, and k components of cell 2 are in the failed state. Note that all three spares must fail before i or k are not zero. Also, if i or k equals two, the system has failed.

With the assumption of a Poisson failure process ([56], page 72) a transition matrix, P, representing the failure process for an interval (t, t + dt) can be developed. Since the quantity of interest is the success probability (reliability) for the system, the transition

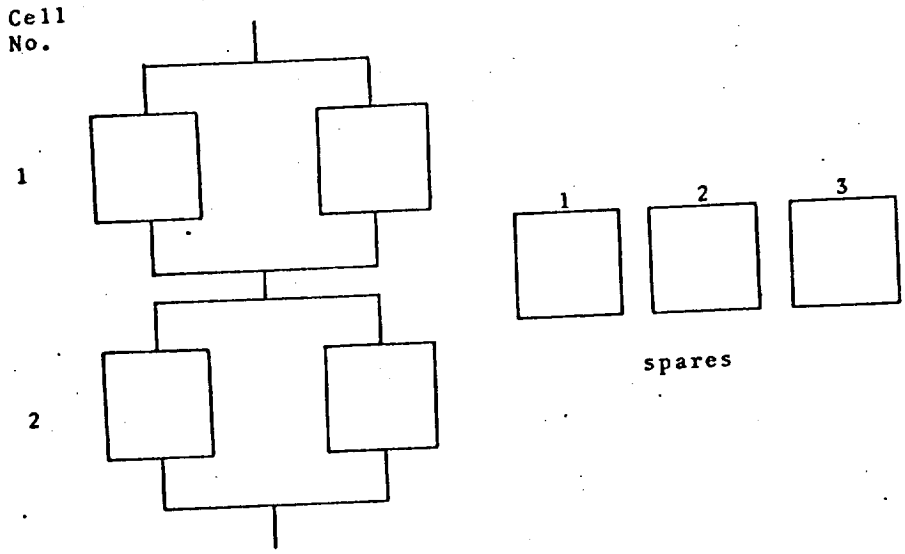


Figure 3.1.--Parallel system with spares, N = 2, m = 3.

matrix for only the success states is presented as:

| | | 000 | 010 | 020 | 030 | 130 | 031 | 131 |
|---|---|---|---|---|---|---|---|---|
| | 000 | $1-4\lambda$ | $4\lambda$ | 0 | 0 | 0 | 0 | 0 |
| | 010 | 0 | $1-4\lambda$ | $4\lambda$ | 0 | 0 | 0 | 0 |
| | 020 | 0 | 0 | $1-4\lambda$ | $4\lambda$ | 0 | 0 | 0 |
| P = | 030 | 0 | 0 | 0 | $1-4\lambda$ | $4\lambda$ | 0 | 0 |
| | 130 | 0 | 0 | 0 | 0 | $1-3\lambda$ | 0 | $2\lambda$ |
| | 031 | 0 | 0 | 0 | 0 | 0 | $1-3\lambda$ | $2\lambda$ |
| | 131 | 0 | 0 | 0 | 0 | 0 | 0 | $1-2\lambda$ |

TABLE 3.1 TRANSITION MATRIX FOR PARALLEL SPARED SYSTEM

where each element in the matrix represents the probability of going from state i at time t to state j at time t + dt. The probability $C\lambda dt$ has been written $C\lambda$ for simplicity.

Note that the probability of being in state j at time t + dt is the sum of the products of the probability of being in each state at time t and the probability of transferring to state j in the interval (t, t + dt). Thus the following equations of detailed balance can be written:

$$
\begin{aligned}
P_{000}(t+dt) &= (1-4\lambda dt)P_{000}(t) \\
P_{010}(t+dt) &= (1-4\lambda dt)P_{010}(t)+4\lambda dtP_{000}(t) \\
P_{020}(t+dt) &= (1-4\lambda dt)P_{020}(t)+4\lambda dtP_{010}(t) \\
P_{030}(t+dt) &= (1-4\lambda dt)P_{030}(t)+4\lambda dtP_{020}(t) \\
P_{130}(t+dt) &= (1-3\lambda dt)P_{130}(t)+2\lambda dtP_{030}(t) \\
P_{031}(t+dt) &= (1-3\lambda dt)P_{031}(t)+2\lambda dtP_{030}(t) \\
P_{131}(t+dt) &= (1-2\lambda dt)P_{131}(t)+2\lambda dtP_{130}(t)+2\lambda dtP_{031}(t).
\end{aligned}
\tag{3.9}
$$

By collecting like terms and dividing by dt for each equation, the differential equations are obtained. For example,

$$
P_{000}'(t) = \frac{P_{000}(t+dt) - P_{000}(t)}{dt} = -4\lambda\, P_{000}(t). \tag{3.10}
$$

Taking the Laplace transform of equation (3.10), and noting that $P_{ijk}(0) = 1$, for i=j=k=0, and $P_{ijk}(0) = 0$, otherwise, the transformed value, $P_{000}(s)$ is obtained:

$$P_{000}(s) = \frac{1}{s + 4\lambda} . \qquad (3.11)$$

Similarly, taking the Laplace transform of each differential equation representing a success state, and solving for P(s), the resulting expressions are:

$$P_{010}(s) = \frac{4\lambda}{(s+4\lambda)^2} , \quad P_{130}(s) = \frac{2\lambda(4\lambda)^3}{(s+4\lambda)^4\ (s+3\lambda)}$$

$$P_{020}(s) = \frac{(4\lambda)^2}{(s+4\lambda)^3} , \quad P_{031}(s) = \frac{2\lambda(4\lambda)^3}{(s+4\lambda)^4\ (s+3\lambda)}$$

$$P_{030}(s) = \frac{(4\lambda)^3}{(s+4\lambda)^4} , \quad P_{131}(s) = 2\ \frac{(2\lambda)^2\ (4\lambda)^3}{(s+4\lambda)^4 (s+3\lambda)(s+2\lambda)} .$$

The above transformed probabilities were developed for m = 1, N = 2, 3, 4, 5, 6 and for m = 2, 3, 4. From the resulting expressions, the following generalizations were induced.

The sum of the set of probabilities relating to the states with only the spares having failed are easily seen to be the Poisson summation with parameter $2N\lambda t$. The transformed generalization for this probability is:

$$\sum_{x=0}^{m} \frac{(2\ N\ \lambda)^x}{(s\ +\ 2\ N\ \lambda)^{x+1}} \ . \qquad (3.12)$$

Taking the inverse Laplace transform ([13], page 295), (3.12) becomes:

$$\sum_{x=0}^{m} \frac{(2\ N\ \lambda\ t)^x}{x!}\ e^{-2N\lambda t} \ . \qquad (3.13)$$

The probabilities for the states corresponding to failed components in the basic parallel system are similarly generalized in the transformed (s domain) as:

$$\sum_{x=1}^{N} \frac{\binom{N}{x}\ (2\lambda)^x\ x!(2N\lambda)^x}{(s+2N\lambda)^{m+1}\ \prod_{j=1}^{x}[s+(2N-j)\lambda]} \ . \qquad (3.14)$$

In order to obtain the inverse Laplace transform of (3.14) note that the numerator is a constant with respect to s and that the inverse transform of the general term of the summation is sought. The approach is to expand a portion of the expression, namely the second term of the denominator, by partial fraction expansion, multiply through by the first term of the denominator, and find the inverse transform of each term of the expansion. By partial fraction expansion, the second term of the

denominator can be written:

$$\frac{1}{\prod_{j=1}^{x}[s+(2N-j)\lambda]} = \sum_{k=1}^{x} \frac{1}{\left[\prod_{\substack{j=1 \\ j\neq k}}^{x}(2N-j)\lambda-(2N-k)\lambda\right][s+(2N-k)\lambda]}, \quad (3.15)$$

where the first term in the denominator represents the partial fraction expansion coefficients. Simplifying the expression (3.15) yields:

$$\lambda^{1-x} \sum_{k=1}^{x} \frac{1}{[s+(2N-k)\lambda]\prod_{\substack{j=1 \\ j\neq k}}^{x}(k-j)}. \quad (3.16)$$

Substituting the expression (3.16) for its equivalent in (3.14) written without the constant in the numerator and for only the general term of the summation, the expression becomes:

$$\frac{\lambda^{1-x}}{(s+2N\lambda)^{m+1}\sum_{k=1}^{x}[s+(2N-k)\lambda]\prod_{\substack{j=1 \\ j\neq k}}^{x}(k-j)}, \quad (3.17)$$

which can be written:

$$\lambda^{1-x} \sum_{k=1}^{x} \frac{1}{(s+2N\lambda)^{m+1}[s+(2N-k)\lambda]\prod_{\substack{j=1 \\ j\neq k}}^{x}(k-j)} \quad (3.18)$$

This expression is such that each term of the summation is of the form:

$$g(s) = \frac{K}{(s+\alpha)^n (s+\beta)}, \tag{3.19}$$

where $\alpha$, $\beta$, and K are constants defined as:

$$\alpha = 2 N \lambda, \tag{3.20}$$

$$\beta = (2 N - k) \lambda, \tag{3.21}$$

$$K = \frac{\lambda^{1-x}}{\prod_{\substack{j=1 \\ j \neq k}}^{x} (k - j)}. \tag{3.22}$$

The inverse transform of expression (3.19) can be determined in the following manner.

Letting $r = s + \beta$ and substituting for s in equation (3.19),

$$g(r) = \frac{K}{(r + \alpha - \beta)^n r}, \tag{3.23}$$

which is the form of a transform given in Roberts and Kaufman [54], page 203. Also from [54], page 4, the expression $g(s + a)$ has the inverse transform $e^{-at} f(t)$. Hence, the inverse transform for equation (3.19) is:

$$f(t) = K e^{-\beta t} (\alpha-\beta)^{-n} \left\{ 1 - e^{-(\alpha-\beta)t} \sum_{y=0}^{n-1} \frac{(\alpha-\beta)^y}{y!} \right\}. \tag{3.24}$$

Substituting in (3.24) the parameters of (3.18) and the value of K given by equations (3.20), (3.21), and (3.22), the inverse Laplace transform of (3.18) is:

$$\lambda^{1-x} \sum_{k=1}^{x} \frac{e^{-(2N-k)\lambda t}}{\prod_{\substack{j=1 \\ j\neq k}}^{x} (k-j)[2N\lambda-(2N-k)\lambda]^{m+1}} \left\{1 - e^{-[2N\lambda-(2N-k)\lambda]t} \sum_{y=0}^{m} \frac{[2N\lambda-(2N-k)\lambda]^{y}}{y!}\right\}. \quad (3.25)$$

Simplifying (3.25) results in the expression:

$$\frac{e^{-2N\lambda t}}{\lambda^{m+x}} \sum_{k=1}^{x} \frac{(-1)^{x+k} \binom{x}{k}}{k^{m}\, x!} \left[e^{k\lambda t} - \sum_{y=0}^{m} \frac{(k\lambda t)^{y}}{y!}\right]. \quad (3.26)$$

Multiplying (3.26) by the numerator given in (3.14) and summing over x, x = 1, 2, ..., N, yields the probabilities corresponding to failed components. Adding this result to the probabilities corresponding to the states with only the spares having failed, (3.13), and simplifying, the expression for the reliability, R(t), for the system of Figure 2.3 for n = 2 is:

$$R(t) = \sum_{x=0}^{m} \frac{(2N\lambda t)^x}{x!} e^{-2N\lambda t} +$$

$$(2N)^m e^{-2N\lambda t} \sum_{x=1}^{n} \binom{N}{x} 2^x \left\{ \sum_{k=1}^{x} \frac{(-1)^{x+k} \binom{x}{k}}{k^m} \left[ e^{k\lambda t} - \sum_{y=0}^{m} \frac{(k\lambda t)^y}{y!} \right] \right\}. \quad (3.27)$$

The reliability of a system having common parallel components with spares is calculated by the product of the K component set reliabilities, (3.27). This product corresponds to equation (3.7), which is the reliability of a single thread system with spares.

Equation (3.27) can be extended for any value of n, but from experience with deriving the differential equations for n = 3 the combinatorics of possible failures becomes extremely tedious. For the application to spacecraft systems, moreover, it is doubtful that more than two components would be put in parallel when the components are replaced upon failure, since the purpose of parallel operation is merely to avoid any system downtime. Hence, for the purpose of presenting the effect of parallel components with spares, the above derivation should suffice.

It should be mentioned that in the process of deriving equation (3.27) several attempts were made to

generalize by transforming each probability back into the t variable from the s domain, and then deduce the general expression. Such an approach in all cases was either unsuccessful or resulted in far more involved algebra than the approach described above. It is recommended that for problems of this type all generalizations be accomplished in the s domain, and then transformed back into the t variable, using algebraic manipulations such as (3.15) when required.

In the following sections pertinent quantitative effects on reliability of the systems described above are presented, and some generalization which influence the system design are stated.

## Comparison of Reliability Values for Spared Systems

The remainder of the chapter illustrates some of the effects on reliability and on the design of the system which should be considered in view of the reliability statements derived above.

In Figures 3.3 to 3.7, the effect on the system reliability of having two components in parallel and backing up with spares (solid lines) is compared with a single thread system with various numbers of spares, m, (dashed lines). The figures represent a different number

of series units of the common component, N, N=1, 2, 3, 5, 10, for Figures 3.3 to 3.7, respectively. The abscissa, $\lambda t$, is the failure rate of a single component of the component set multiplied by the mission duration and the ordinate, R, is the component set reliability. Figure 3.2 is a pictorial representation of the component sets of Figures 3.3 to 3.7. Component Set A has the simple series system as its basic components and Component Set B the series-parallel set.

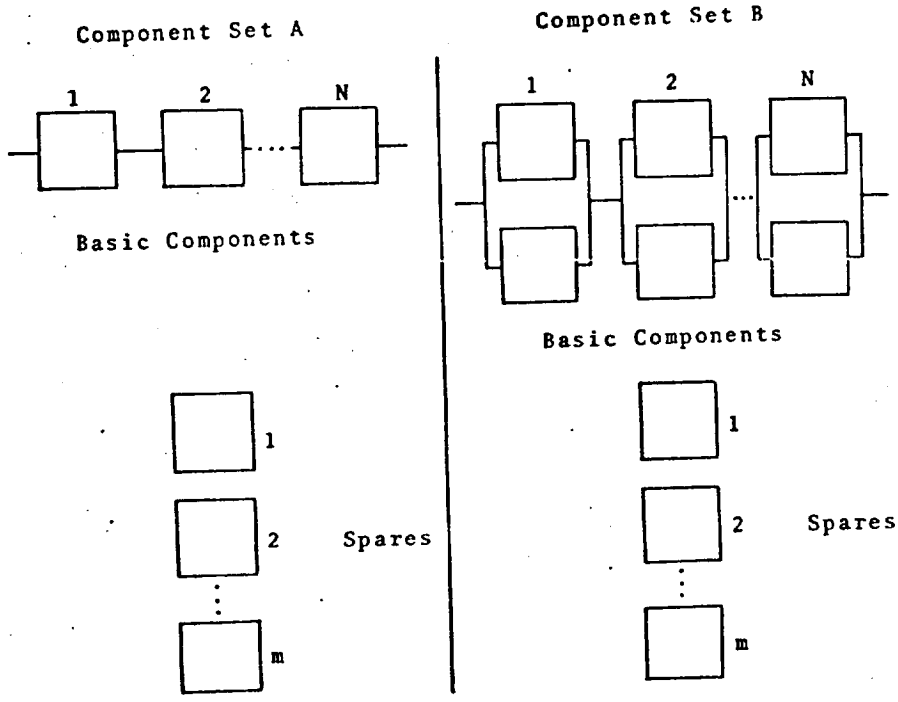


Figure 3.2.--Representation of component sets of figures 3.3 to 3.7.

It is of interest to note the following characteristics of these curves. In Figure 3.3 (N = 1), for m = 2, note that a system consisting of a single component with two spares has a higher reliability than two components in parallel with two spares for $\lambda t > .9$. These two systems are represented in Figure 3.2 for N = 1, where the above observation is that the reliability of component set A is greater than that of component set B for $\lambda t > .9$. This fact, extended to other values of N, m, and $\lambda t$ should be of interest in the design of redundant systems, and will be elaborated upon in the following sections.

In Figure 3.4, it is noted that one spare with a single thread system (N = 2) has an almost identical reliability to that of a system with two series parallel components and no spares, for any value of $\lambda t$ in the range of the graph. The latter component set has four components and the former a total of three components, including the spare. From observing the similarity between these curves for the series and series-parallel system, it is suggested that one curve may be used to approximate the other. Experience with computing the equation (3.27) on the IBM 360/65 has shown that such an approximation may indeed be required if the series-parallel system is used in an optimization algorithm. For example, to compute equation (3.27) for M = 1, ..., 12 and N = 1, ..., 20, approximately

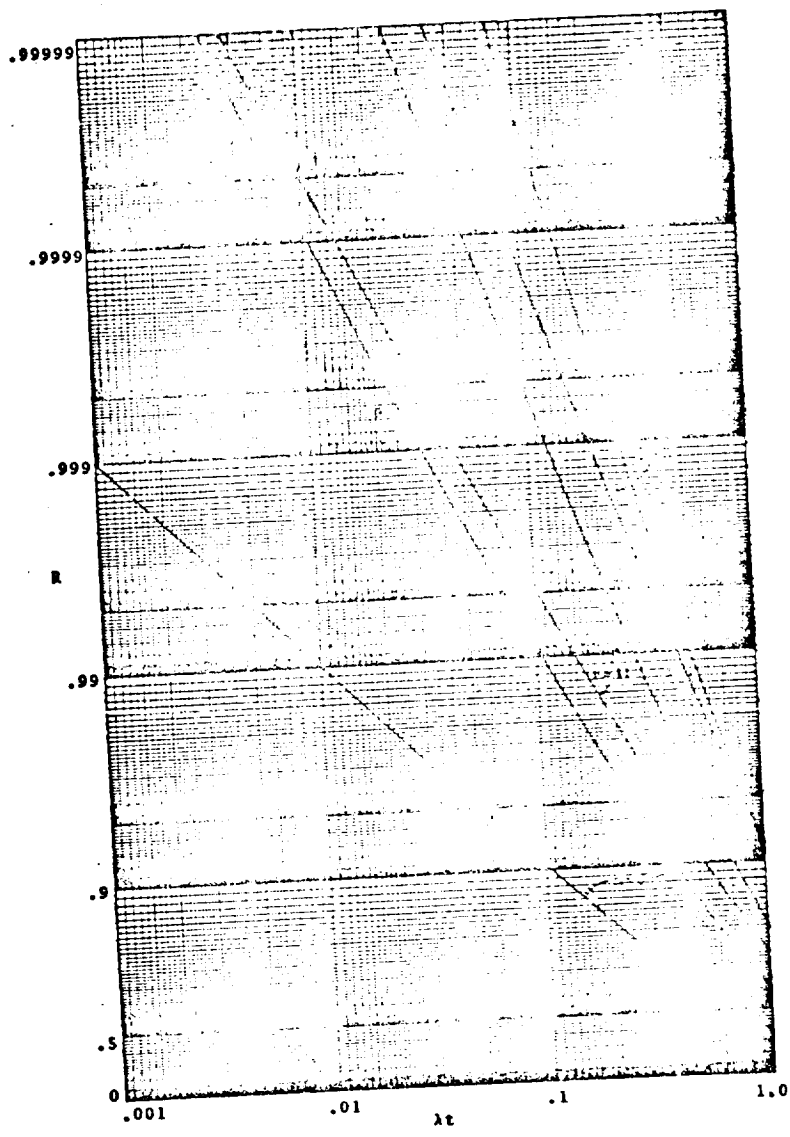


Figure 3.3.--Reliability vs λt for series-parallel and series systems with spares, illustrated in Figure 3.2, N=1.

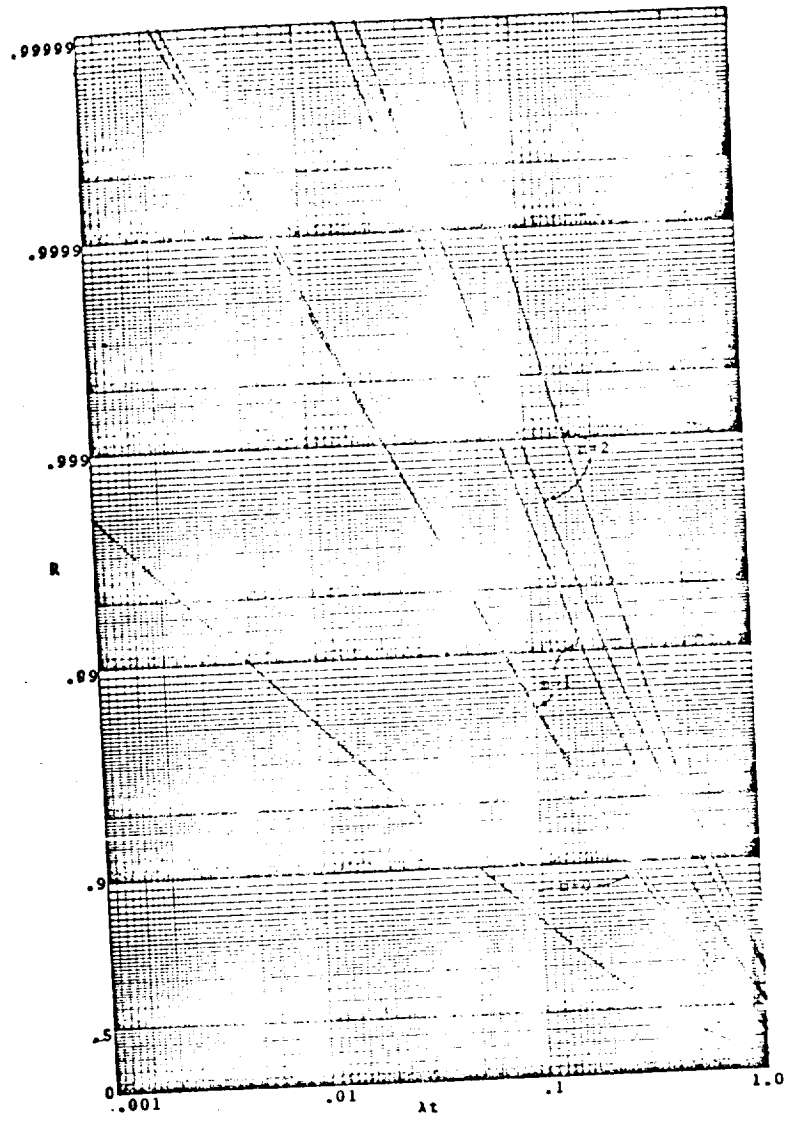


Figure 3.4.--Reliability vs λt for series-parallel and series systems with spares, illustrated in Figure 3.2, N=2.

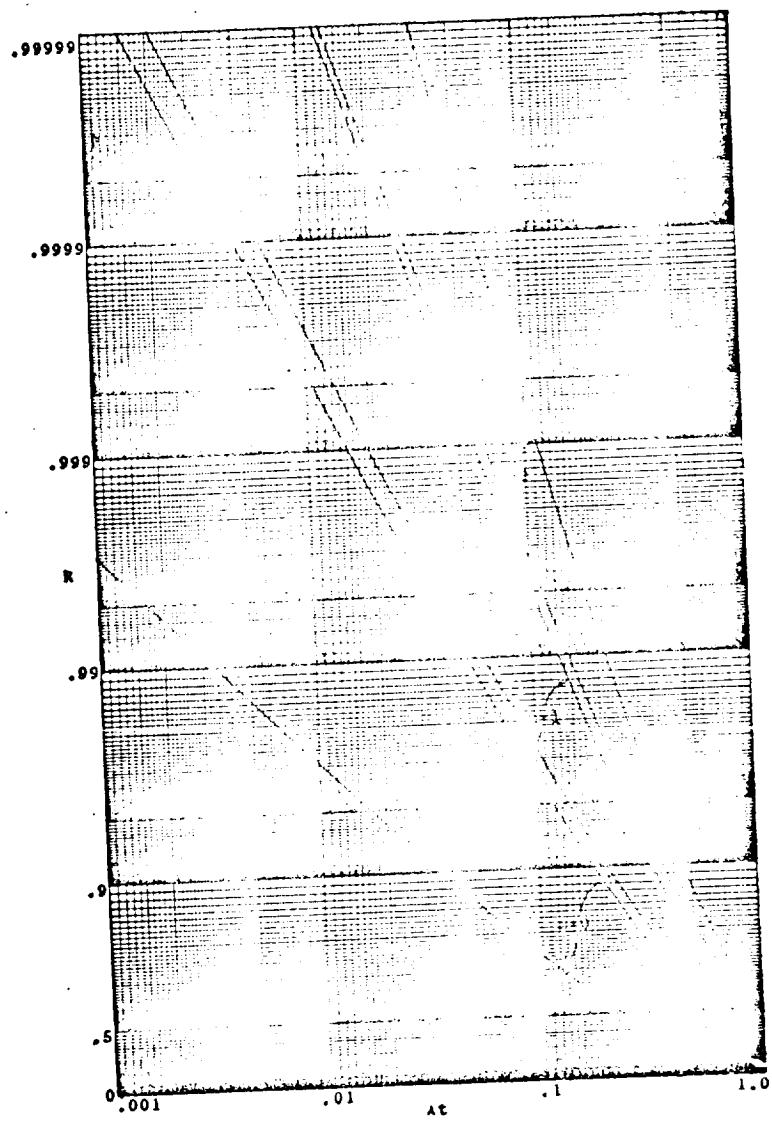


Figure 3.5.--Reliability vs λt for series-parallel and series systems with spares, illustrated in Figure 3.2, N=3.

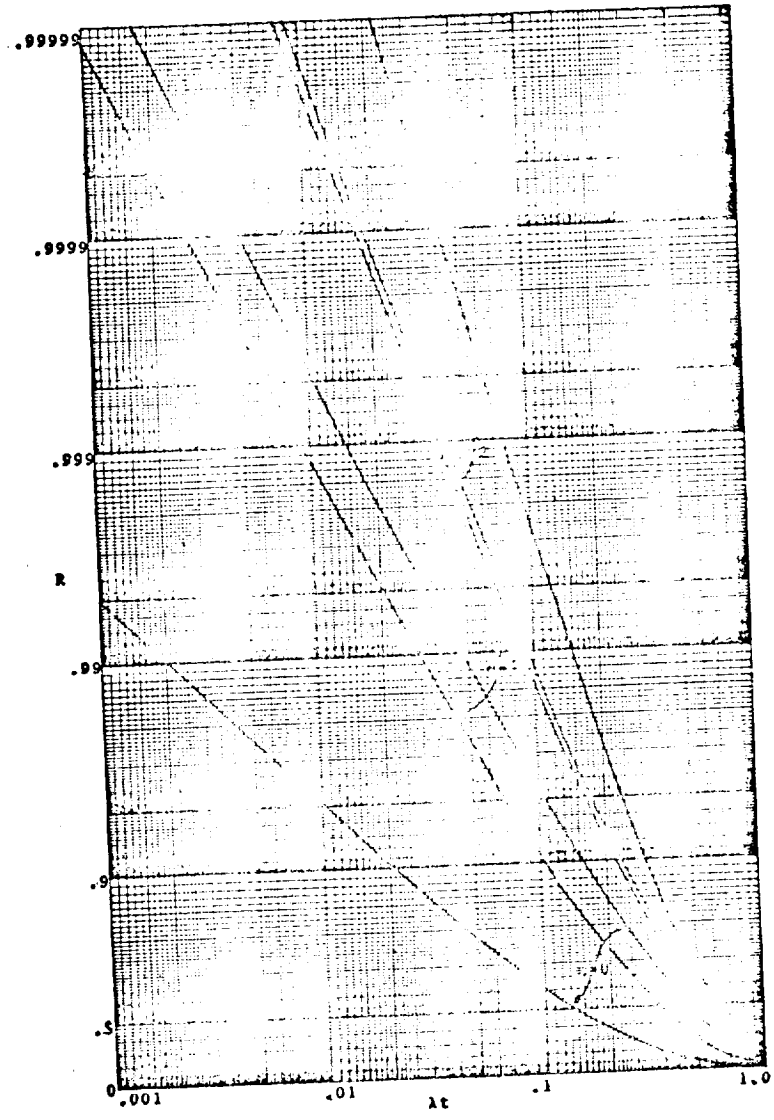


Figure 3.6.--Reliability vs λt for series-parallel and series systems with spares, illustrated in Figure 3.2, N=5.

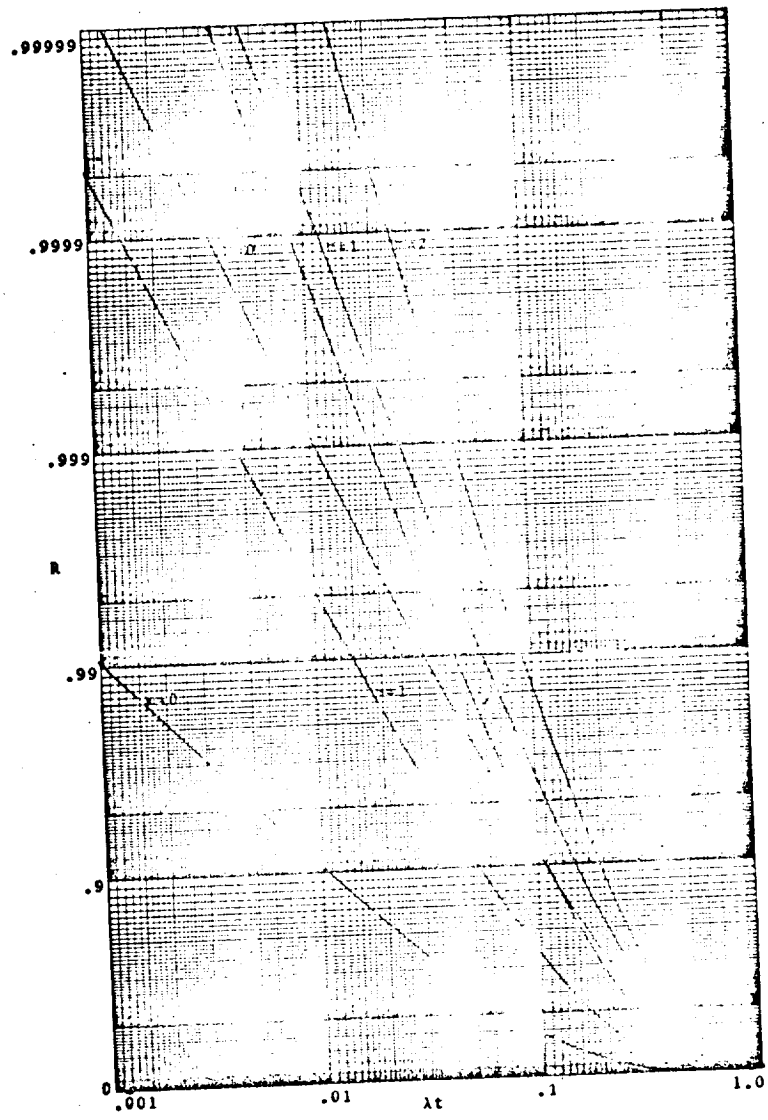


Figure 3.7.--Reliability vs λt for series-parallel and series systems with spares, illustrated in Figure 3.2, N=10.

two minutes of computer time are required. Specific recommendations for approximation are given in Chapter VII.

The next section will discuss and demonstrate graphically some of the effects of commonality in spared systems.

## Effects of Commonality

In studies which investigate the use of redundancy to obtain the required reliability for long space missions [42], it is mentioned that there is an advantage to having common components in the system when possible. However, no quantification of the effects of having commonality is attempted in the literature reviewed. In this section, an attempt will be made to convey some notion of the extent to which commonality of components affects the reliability and weight of a spared system.

In Figure 3.8 a comparison is made between the spares required for a system consisting of N like components, N = 1, 2, ..., 100, and N components all of the same weight, but not interchangeable. The value of $\lambda t$ for the component used for this example is .10, which results in a single component reliability of approximately .99. The system reliability is required to be greater than or equal to .99.

As N gets large, the difference between the spares

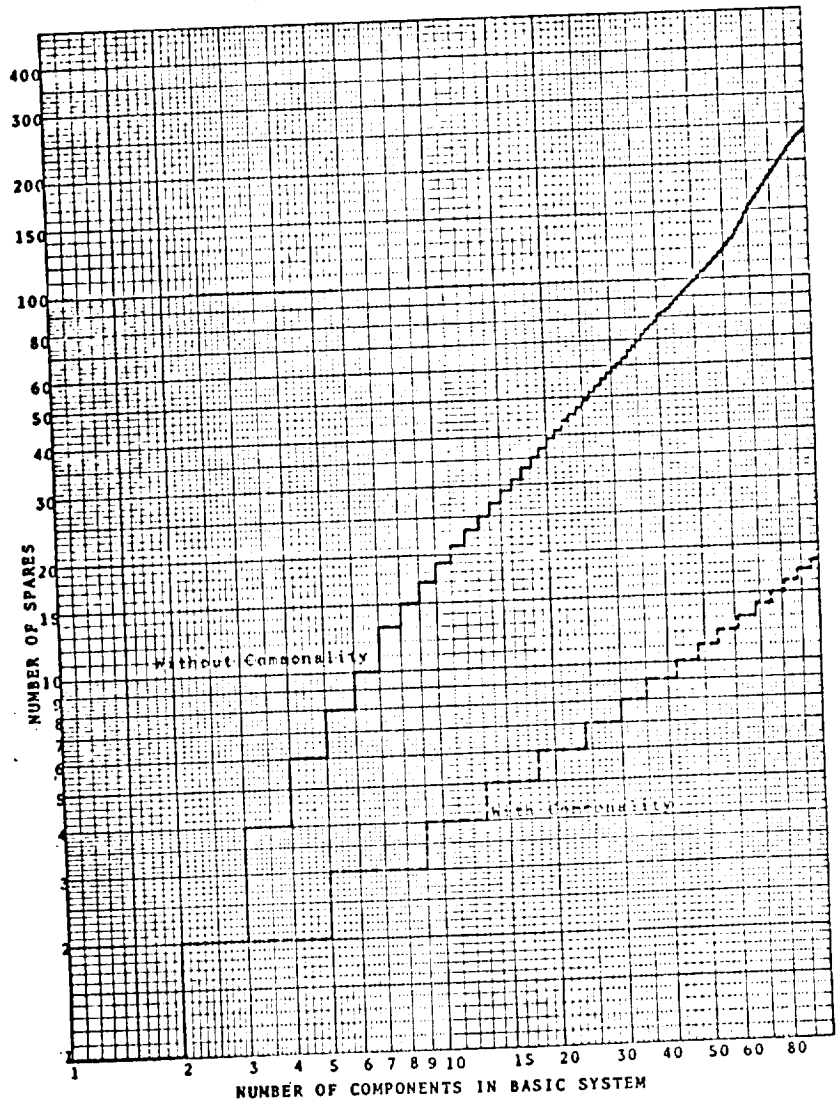


Figure 3.8.--Effect of commonality on spares required for various numbers of components in system, $\lambda t = .10$, $R \geq .99$.

required (and thus the total weight) of the system is very marked. For example, for N = 50, 11 spares are required for the system with all components interchangeable, while 100 components are required for the system with no commonality.

To illustrate some of the design trade-offs which may be made to take advantage of commonality for long space missions, the following example is presented and illustrated in Figure 3.9. Suppose, in the comparison made in Figure 3.9, it is possible to design certain components to serve the same function in the spacecraft, but only at the expense of an increase in the weight of the component. A question which may be posed, then, is: For what increase in component weight does the system of common components have a lower weight than the no-commonality alternative? Figure 3.9 compares these two alternatives under the assumption that the individual component weight is increased by 50 percent in order to achieve commonality, for $\lambda t = .1$. The weight shown is the total weight including the original system and spares in units of components. Even under this assumption, the system with all components interchangeable has a lower weight for $N > 3$.

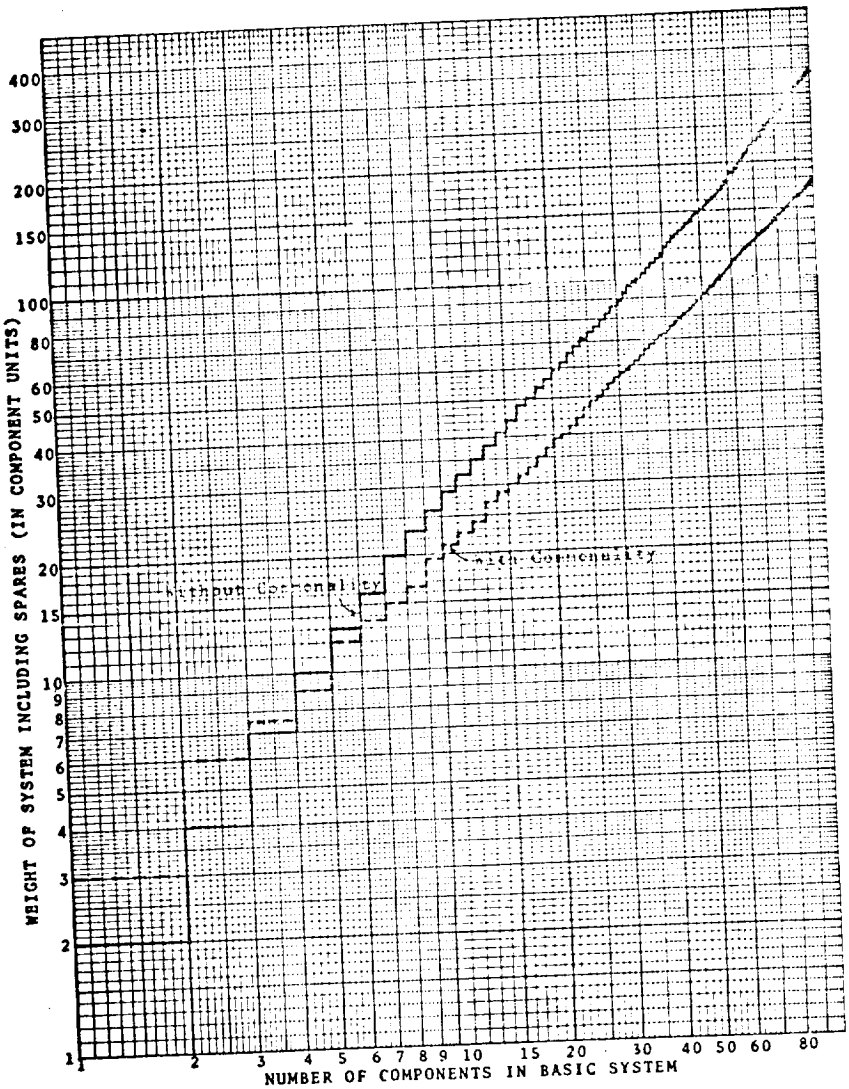


Figure 3.9.--Effect of commonality on spares required with module weight increased 50% for commonality case, $\lambda t = .10$, $R \geq .99$.

## Design Generalizations

The previous sections have illustrated some specific results of having components in parallel in the basic system with spares and of having identical components in a system with spares. In this section, some generalizations of these results will be stated.

It was noted in Figure 3.3 that for m = 2 the particular systems illustrated in Figure 3.2 had reliability versus $\lambda t$ curves which cross at $\lambda t \doteq .9$. A generalization of this result is that for each value of $\lambda t$, there will be some value of n, the number of components in parallel, and some value of m, the number of spares, for which the reliability with n + 1 components in parallel is less than the reliability with n components in parallel for the same value of m. A pictorial representation of the system of this discussion is shown in Figure 3.10. Further, the reliability will not continue to decrease with increasing n, but will, for some n, reach a minimum value and then increase monotonically, approaching 1.0 as a limit. Figures 3.11 and 3.12 illustrate the reliability, R, of the system of Figure 3.10, with $\lambda t$ = .5 and 1.0, respectively.

The values of m and n are incremented to sufficiently large values to illustrate a minimum point of the

reliability versus n curve for some m.

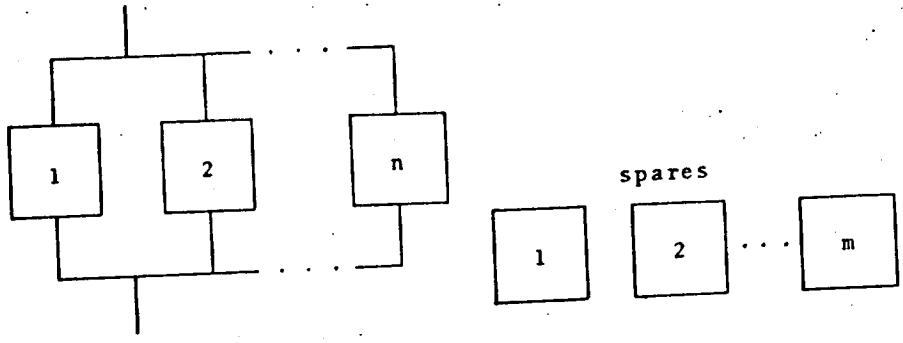


Figure 3.10.--Representation of a system with n parallel components and m spares.

For design purposes a rather important generalization should be realized from the above discussion. Namely, that addition of an active parallel component in systems with spares may not only result in significantly less increase in reliability than the addition of a spare, but may actually reduce the system reliability. Hence, under the assumptions of this research, redundancy in the form of active parallel components should only be used when system downtime cannot be allowed while replacement is being accomplished or when it is infeasible to replace failed components.

For m = 0, the reliability versus n curves are

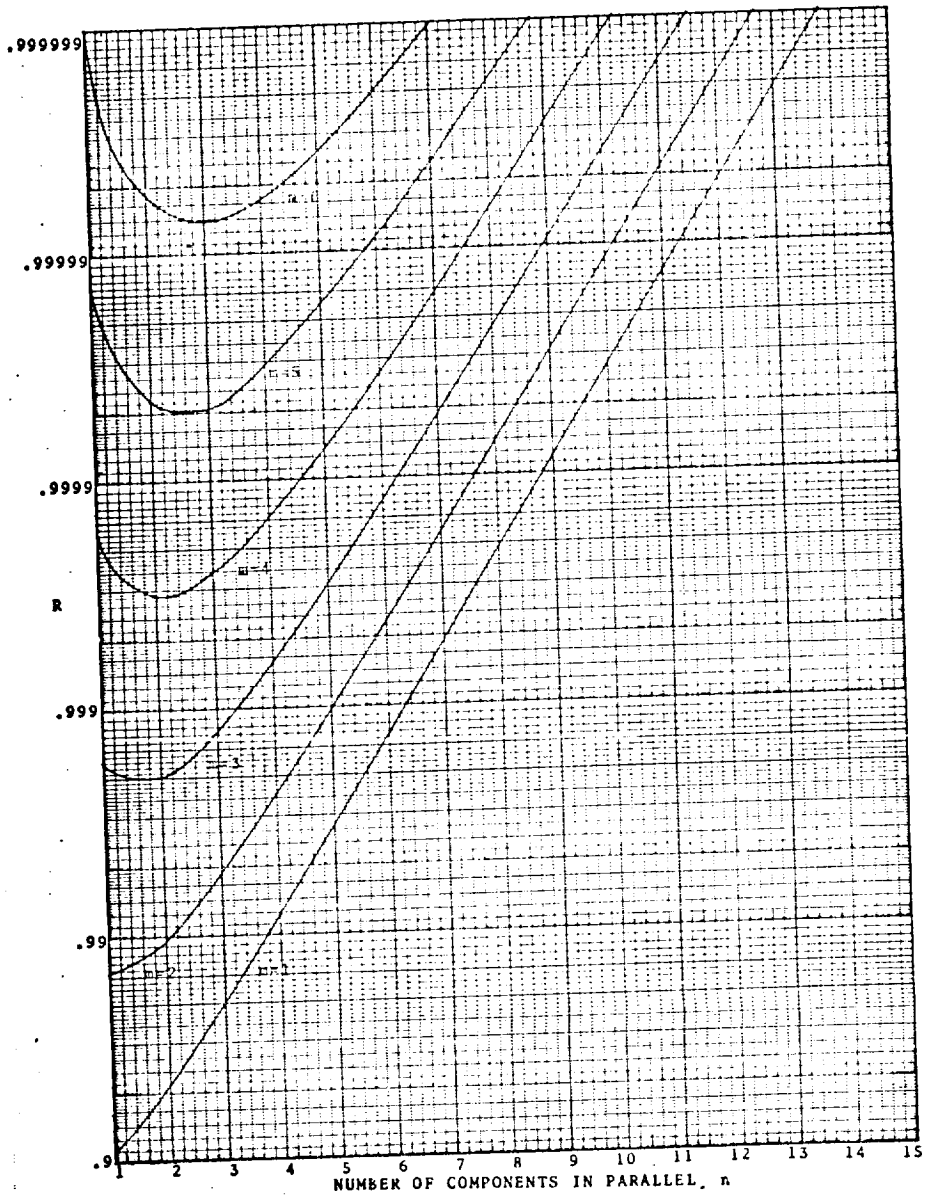


Figure 3.11.--Reliability vs number of components in parallel, $\lambda t = .5$.

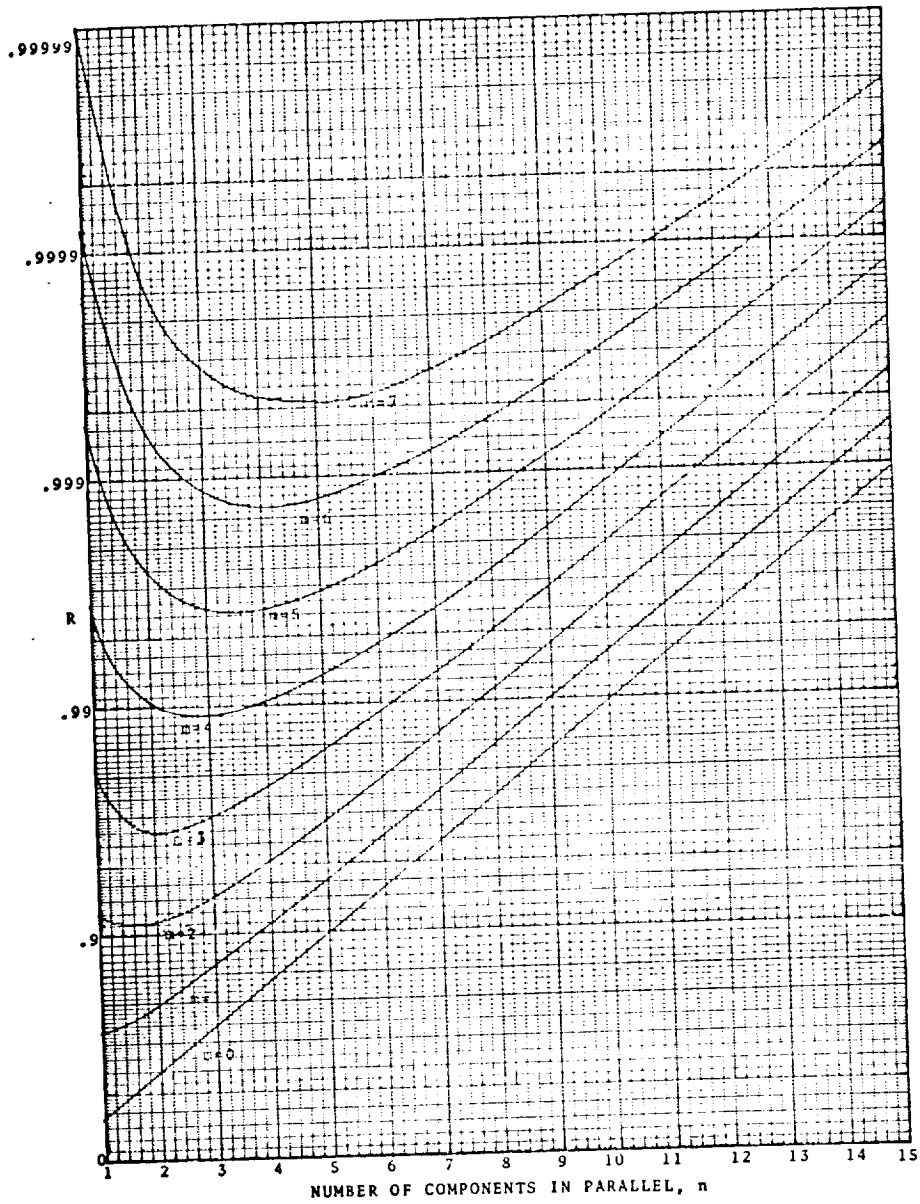


Figure 3.12.--Reliability vs number of components in parallel, $\lambda t=1.0$.

monotonic increasing in n for any m (number of spares) and for any value of $\lambda t$. A trade-off may be effected between adding a number of components in parallel and designing the system for replacement with spares. For example, from Figure 3.12 it is noted that approximately the same reliability is obtained for 12 components in parallel with no spares as for only 4 spares with no parallel components (n = 1). The effect of using spares compared with components in parallel is accentuated when a set of components in series is considered, as illustrated in Figure 3.8.

# CHAPTER IV

## FORMULATION OF THE OPTIMUM MODULE SIZE PROBLEM

The optimum module size problem was generally stated in the Introduction. In this chapter, the problem is stated mathematically, the specific set of assumptions used in solving the problem are outlined, data requirements are discussed, and the problem is formulated as an integer programming problem.

Reference to Figure 4.1 and Table 4.1 may be helpful in clarifying the terms used in the following problem definition. Table 4.1 presents a set of values for the example system illustrated in Figure 4.1.

The spacecraft is considered as having been designed L different times, each time with provision for replacement of failed components at the $j^{th}$ level, $j = 1,2,\ldots,L$. For the system represented in Figure 4.1, $L = 3$. Each level of component replacement is simply an aggregation of components of the next lower level, beginning with the lowest (smallest parts) level at which it is feasible to replace failed components. All components of the spacecraft design selected are not required to be at the same level.

The basic question concerns the modular level,

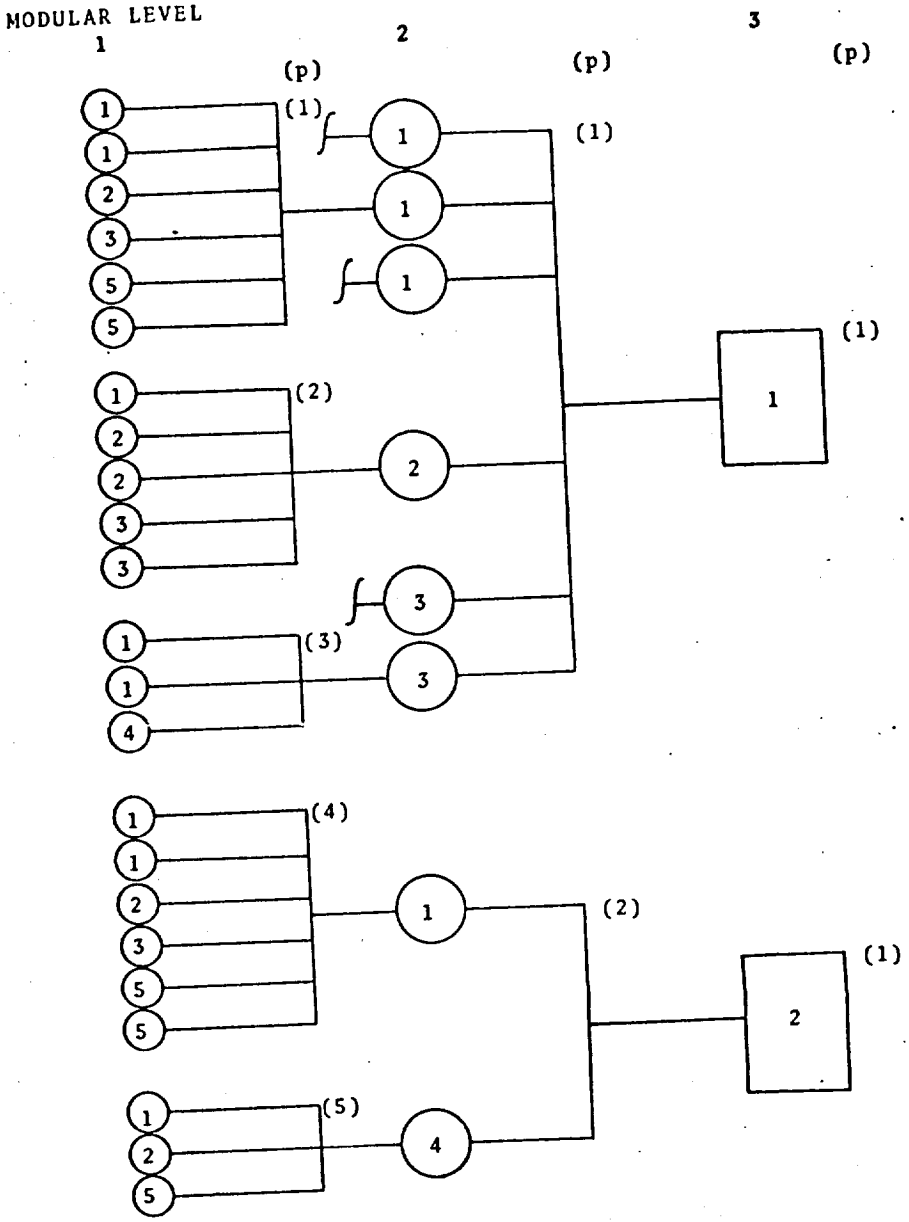


Figure 4.1.--Example system of candidate modules.

considering replacement, at which to design the spacecraft in order to minimize the total weight, including spares, required for a given mission. It will become clear that the solution to the optimum module size requires a simultaneous solution to the optimum number of spares to carry on a given mission.

First the terms required for stating the objective function, to minimize the total spacecraft weight, are defined. In the following sections the factors included in spacecraft weight are discussed.

Let $w_{ij}$ = weight of the $i^{th}$ module at the $j^{th}$ modular level, $i=1,2,\ldots,N_j$, $j=1,2,\ldots,L$, where $N_j$ is the number of different types of modules at level j.

$m_{ij}$ = number of spares of the $i^{th}$ module at the $j^{th}$ level, $m_{ij}=1,2,\ldots,M_{ij}$, where $M_{ij}$ is an upper limit on the number of spares of module i at level j which is permitted.

The number of spares for each module at each level, $m_{ij}$, is a decision variable. A second decision variable is required to specify which of the candidate modules to include in the design of the basic system.

The number of spares, $m_{ij}$, relates only to a specific module (i,j), since a spare can replace a failed module of that type in any location in the basic configuration.

However, the decision variable specifying which modules to include in the design must also identify which specific location of the design hierarchy each module type occupies. This requirement arises because provision must be made for sparing a given module for one location, while in another location, the same module is combined with others into a higher level replacement module, precluding its replacement in the latter location. For example, in Figure 4.1 there are 14 candidate positions for module (i,j)=(1,1) in the basic system (including those in the repeated sets of components indicated by truncated connecting lines). However, if module (3,2) is to be included in the basic system, 4 of the locations of module (1,1) are eliminated, since replacement within the replacement module (3,2) is not allowed.

The identification of the location of a module (i,j) must be unique with respect to the different modules into which it can be integrated in the higher levels of the hierarchy. This identification will be called the "path", denoted by (p). Distinguishing between locations of modules having the same set of higher level modules is not required, as discussed in the following section.

To illustrate the term "path", consider the module (i,j)=(1,1) in Figure 4.1. This module has 5 paths, since there are 5 unique paths which can be traced through the

hierarchy to the highest level, level 3. Note that for $p = 1$, $p = 4$, and $(i,j)=(1,1)$, the paths do not differ at $j = 2$, but are nevertheless unique since each leads to a different module at $j = 3$. There are only two paths from $j = 2$, since there are only two higher level components to which each path can lead.

Using the above definition of path, let

$k_{ij}^{(p)}$ = the number of uses of the $i^{th}$ module at the $j^{th}$ level for path p, p=1, 2, ..., $P_j$, where $P_j$ is the number of paths at level j.

For example, in Figure 4.1, $k_{11}^{(1)} = 6$. Note that the truncation of a connecting line of the hierarchy represents a repetition of a portion of the modular levels illustrated elsewhere in the same path. Table 4.1 gives each of the values of $k_{ij}^{(p)}$ for the example illustrated in Figure 4.1.

The second decision variable determines whether or not a given module appears in the basic design and is defined as:

$$\delta_{ij}^{(p)} = \begin{cases} 1, & i, j, p \in U \\ 0, & i, j, p \notin U \end{cases},$$

where U is the set of modules used in the spacecraft design.

The objective function can be stated as:

| i \ j | p | 1 | 2 | 3 | 4 | 5 | 2 | | 3 |
|---|---|---|---|---|---|---|---|---|---|
| | | 1 | | | | | 1 | 2 | 1 |
| 1 | | 6 | 1 | 4 | 2 | 1 | 3 | 1 | 1 |
| 2 | | 3 | 2 | 0 | 1 | 1 | 1 | 0 | 1 |
| 3 | | 3 | 2 | 0 | 1 | 0 | 2 | 0 | |
| 4 | | 0 | 0 | 2 | 0 | 0 | 0 | 1 | |
| 5 | | 6 | 0 | 0 | 2 | 1 | | | |

TABLE 4.1 VALUES OF $k_{ij}^{(p)}$ FOR FIGURE 4.1

$$\min_{m_{ij},\delta_{ij}^{(p)}} Z(m_{ij},\delta_{ij}^{(p)}) = \left\{ \sum_{j=1}^{L} \sum_{i=1}^{N_j} w_{ij} \left( m_{ij} + \sum_{p=1}^{P_j} \delta_{ij}^{(p)} k_{ij}^{(p)} \right) \right\} \quad . \quad (4.1)$$

In order for the values of $m_{ij}$ and $\delta_{ij}^{(p)}$ selected in minimizing (4.1) to be feasible two basic requirements must be satisfied.

1. A spacecraft design having at least one of each of its required single thread† modules must result.
2. A specified reliability requirement must be satisfied.

Feasibility requirement (1) is accomplished by the proper selection of the $\delta_{ij}^{(p)}$ variables. For example, in

†A single thread system is one which contains the minimum number of components required to perform the system function.

Figure 4.1 a feasible selection of the $\delta_{ij}^{(p)}$ is:

$$\delta_{ij}^{(p)} = \begin{cases} 1, & i=1, \ldots, 5,\ p=1,\ldots,5,\ j=1, \\ 0, & \text{otherwise.} \end{cases}$$

Another feasible solution in terms of the $\delta_{ij}^{(p)}$ for Figure 4.1 is:

$$\delta_{ij}^{(p)} = \begin{cases} 1, & i=1,2,3,5,\ p=1,2,4,5,\ j=1, \\ 1, & i=3,\ p=1,\ j=2, \\ 0, & \text{otherwise.} \end{cases}$$

A restriction on those values of $\delta_{ij}^{(p)}$ which may be given a value of 1 is that the components of a replacement module cannot be replaced in that module for the same p (but may be replaced elsewhere). For example, in Figure 4.1 a solution which has $\delta_{11}^{(1)} = 1$ and $\delta_{12}^{(1)} = 1$ is infeasible. However, a feasible solution may have $\delta_{12}^{(1)} = 1$ and $\delta_{11}^{(4)} = 1$.

It may be noted that requiring at least one of each set of single thread modules to exist in the design resulting from a set of values of the $\delta_{ij}^{(p)}$ is equivalent to requiring exactly one set of the modules required for a single thread system. Any excess modules would increase the value of the objective function and decrease the reliability, automatically eliminating these solutions from consideration. In the subsequent formulation of the problem by integer programming, feasibility requirement (1)

will be stated mathematically.

The second feasibility requirement, that of requiring a specified reliability, is stated mathematically below. The time independent assumption discussed in Chapter III is used to simplify the discussion.

In addition to the symbols defined above, let:

$p_{ij}$ = the probability of failure during a mission of the $i^{th}$ module at the $j^{th}$ level,

$q_{ij} = 1 - p_{ij}$.

$R$ = the desired spacecraft reliability (probability of success).

The reliability constraint, as discussed in the derivation of equations (3.1) and (3.2) is:

$$\prod_{i,j,p\in U} \left[ \sum_{r=0}^{m_{ij}} \binom{K_{ij}^{(p)} + m_{ij}}{r} p_{ij}^{r} q_{ij}^{K_{ij}^{(p)} + m_{ij} - r} \right] \geq R, \qquad (4.2)$$

where $K_{ij}^{(p)} = \sum_{p=1}^{P_j} \delta_{ij}^{(p)} k_{ij}^{(p)}$, $\quad i, j, p \in U.$

An additional condition which must be placed on the constraint (4.2) is that for each i, j such that $K_{ij}^{(p)} = 0$, $m_{ij}$ must also be 0. This simply means that spares cannot be provided for a module (i,j) if no units of that type module are used in the system. From this condition, it

follows that:

$$\sum_{r=0}^{m_{ij}} \binom{K_{ij}^{(p)} + m_{ij}}{r} p_{ij}^{r} q_{ij}^{K_{ij}^{(p)} + m_{ij} - r} = 1.0, \qquad (4.3)$$

for $K_{ij}^{(p)} = 0$, which results in the proper value of constraint (4.2) for this condition.

## Assumptions and Considerations

Before proceeding to a description of the specific mathematical formulation which will be used to develop the algorithm of Chapter V, the specific set of assumptions and considerations of this model are listed. The general set of assumptions, stated in the Introduction, are not repeated here, but are elaborated upon and supplemented.

First, the objective function stated mathematically in the previous section will be re-examined. Generally stated, the objective is to minimize the total weight of the spacecraft. The weight of the spacecraft is considered to be composed of the following elements. For the third element, which is not given as weight, a conversion to weight is given.

1. Weight of the basic module - It should be noted that the sum of the weights of the component parts of a module, if replaced separately, will

invariably exceed the weight of the module if it is considered as the replacement unit.

2. Weight of test equipment (for fault isolation) - It is assumed that the fault isolation equipment is relatively automatic and that it is required only to determine the replacement module which must be replaced to correct a malfunction. Equipment necessary to determine whether subsystems are operating properly is assumed to be the same regardless of replacement module size.

3. Replacement time - The time required to replace a given module includes fault isolation time as well as actual replacement time. It is assumed that any maintenance (replacement of failed modules) work performed by the spacecraft crew will add to the manpower requirement for the mission. The replacement time in man-hours will be converted to weight using the following transformation:

$$r_{ij} = (S/H)h_{ij},$$

where $r_{ij}$ = weight increase due to replacement of module i at level j,

$S$ = total weight that spacecraft must be increased to sustain one man,

$H$ = number of man-hours for the mission that one man is available for maintenance,

$h_{ij}$ = number of man-hours required to replace module i at level j.

It is implicit in the above discussion that all maintenance man-hour requirements will be provided for by adding additional crew members for performing maintenance. The spacecraft weight, and thus the number of maintenance man-hours, given by the optimum value of the objective function (4.1) will be sufficient to replace all of the spares carried on a mission. Since the probability of having to replace all of the spares is extremely remote, provision for using the basic spacecraft crew with a given probability may be desirable to reduce the spacecraft weight.

Although only the total weight requirements of the replacement man-hours are provided for in the objective function, it is assumed that the instantaneous loading effects of simultaneous failures can be handled by the crew. For the purpose of calculating the reliability of the spacecraft with spares, it is also assumed that replacement of malfunctioned modules is instantaneous. The validity of results obtained under these assumptions is supported by [42], in which simulation was used to analyze the time effects of the type of maintenance

activity required for replacement of failed spacecraft systems with spares.

In order to accomplish the required replacement activities at each level, a certain amount of volume will be required to obtain the necessary accessibility to failed modules. It will be assumed that this volume requirement and the attendant weight addition are taken into account in the stated module weights. It is also assumed that the skills required to perform fault isolation and module replacement can be provided by crew training as required. In a similar manner, any common test equipment and tools needed to perform replacement actions are assumed to be apportioned among the modules of a given level if they affect the decision as to the optimum module size. Any system of parts location and storage is also assumed to be apportioned among the various modules when such considerations affect the module size decisions.

In addition to the parameters used in the objective function, the reliability of each candidate replacement module must be estimated so that the total spacecraft reliability can be assessed. It is assumed that the reliability of a module will be greater than the reliability computed by the product of the reliabilities of each of the candidate replacement modules which would substitute for the module if they were chosen as the

modules of the basic system. Although the replacement action of inserting a spare to replace a malfunctioned module is not considered as a detriment to the spacecraft reliability, it is proposed that the reliability of the replacement action can be considered in the stated reliability of each module.

Several of the reliability expressions derived in Chapter III are applicable and usable for the optimum module size problem. Expression (3.8) was chosen for use as the basis for this discussion. Other expressions and the attendant assumptions will be discussed in Chapter V under Variants of the Algorithm. The reliability statement (3.8) assumes that each module i at level j has the same failure rate, $\lambda_{ij}$, where $\lambda_{ij}$ is the parameter of the exponential failure distribution discussed in Chapter III.

In the literature review, it was noted that certain of the references consider a subjective "criticality factor" or a similar method to account for the relative essentiality of a component to the mission. In this research, the effect of each failure is assumed to be equally critical. Put another way, the assumption is made that if any of the single thread components fail and are not replaced, the system fails. Since each part of the basic system is assumed to be essential, allowance is not made for substituting one part of the basic system for

another when there are no spares remaining.

Another type of substitution of basic parts involves the salvaging of the usable parts of failed modules. This type of activity will also not be allowed within the model framework. In the Introduction it was stated that bench repair will not be considered as a basic study guideline. It is also assumed that in determining the parameters of weight and reliability for the candidate replacement modules, the modules at each level are designed for minimum weight and maximum reliability, thus disallowing accessibility to the components within each module.

Also, within the guidelines of considering only the simple maintenance actions of replacement of failed modules, consideration is not given to scheduled maintenance. Scheduled maintenance could be analyzed from the standpoint of tasks not involving replacement of modules with spares, with the result that additional crew time would be necessary to perform a given mission, but without any effect on the optimum module size.

Although it is not essential to the development of the algorithm in Chapter V, the assumption is made that if a given module is selected for the spacecraft design, all modules of that type must be selected for the same path, p, as illustrated in Figure 4.1. It is intuitive that such a solution would result in any case, but some

notational simplification and reduced problem dimensionality is accomplished by stating this condition as an assumption.

Some additional assumptions and guidelines are listed below:

1. Failures are statistically independent.
2. The failure rate of a module is independent of the number of spares for that module carried on a mission.
3. A spare installed to replace a failed module has the same reliability as the initial equipment.
4. None of the spacecraft systems are used as a back-up for another system, since each of the modules of the single thread system is essential.
5. Replacement action performed on one module does not affect the failure of another.
6. Failure rates of modules are known and do not change during a mission.

## Approaches to the Model

At the beginning of this chapter, the optimum module size problem was stated mathematically. However, it was noted that certain requirements were lacking in the mathematical statement, and were accounted for by stating

simply: i, j, p ε U, where i specifies a module at level j, path p, and U is the set of modules used in the design. In this section, the set U will be defined by the mathematical formulation itself.

In order to promote an understanding of the nature of the problem, two formulations will be presented. The first is identical to the second except that there is no commonality of components in the system. This somewhat restricted system leads to a rather straightforward model and analysis which will be presented in its entirety in this section. The second formulation, with commonality, will be discussed as it relates to the no-commonality case, and a complete formulation will be made in the following section.

Without common components.

If there are no identical components in the single thread system, that is each spare is to be used in a unique position, the problem of determining the optimum level at which to perform replacement actions is greatly simplified. In addition to the case of a simple autonomous system, this type of system may appear as a segment of the spacecraft configuration. If a segment of the system has no components in common with the rest of the system and no common components within that segment, the method described in this section may be used to find the optimum for all

values of reliability which can appear in the optimal solution, and thus allow decomposition of the original problem. This approach will be discussed in Chapter V in the section Variants of the Algorithm.

The notation used for the formulation of the optimum module size problem without common components is simpler than that required for the commonality case presented earlier in the chapter, and is listed completely here, omitting the discussion given earlier.

Let $w_{ij}$ = the weight of the $i^{th}$ module at the $j^{th}$ modular level, $i=1,2,\ldots,N_j$, $j=1,2,\ldots,L$.

$m_{ij}$ = the number of spares of the $i^{th}$ module at the $j^{th}$ level, $m_{ij}=1,2,\ldots,M_{ij}$.

$$\delta_{ij} = \begin{cases} 1, & i,j \in U, \\ 0, & i,j \notin U, \end{cases}$$ U is the set of modules used in the spacecraft design,

$\lambda_{ij}$ = the failure rate of the $i^{th}$ component at the $j^{th}$ level.

The objective function is:

$$\min_{m_{ij},\delta_{ij}} \left\{ \sum_{j=1}^{L} \sum_{i=1}^{N_j} w_{ij}(m_{ij} + \delta_{ij}) \right\}. \qquad (4.4)$$

The reliability constraint, under the assumption of exponential failure distribution as discussed in deriving expression (3.7), is:

$$\prod_{i,j\in U}\left[\sum_{r=0}^{m_{ij}} \frac{(\lambda_{ij}t)^r}{r!} e^{-\lambda_{ij}t}\right] \geq R, \qquad (4.5)$$

where R is the required reliability for a mission of length t.

In the case without common components, the set U will be determined by the method of selection of the variables $\delta_{ij}$. The solution technique to be used here is dynamic programming. The dynamic programming formulation and solution will be presented by using the example system illustrated in Figure 4.2.

Table 4.2 gives a listing of the stages of dynamic programming, which are simply the modules selected in an appropriate order.

| Stage | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Module | i | 1 | 2 | 3 | 4 | 1 | 5 | 6 | 7 | 2 | 8 | 9 |
| | j | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 1 |

| Stage | | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Module | i | 3 | 1 | 10 | 11 | 12 | 4 | 13 | 14 | 5 | 2 |
| | j | 2 | 3 | 1 | 1 | 1 | 2 | 1 | 1 | 2 | 3 |

TABLE 4.2 DYNAMIC PROGRAMMING STAGES FOR THE EXAMPLE PROBLEM OF FIGURE 4.2

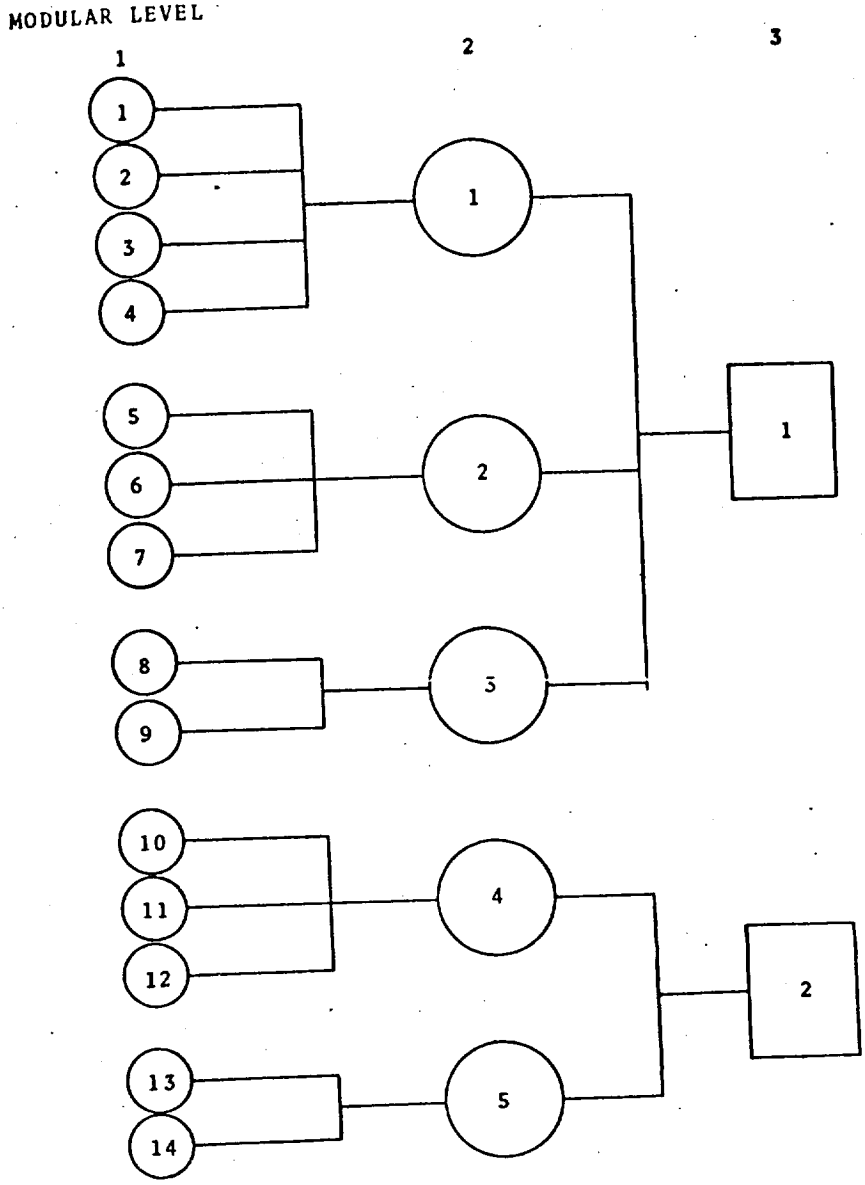


Figure 4.2.--Example system of candidate modules without commonality.

It has been shown [57] that when a problem having a product constraint with a sum objective is to be solved using dynamic programming, the dual of the problem, maximizing the product function subject to the sum constraint, is advantageous. By solving the dual, discrete approximation of the state space, in this case reliability, is obviated. Hence, the problem which is solved using dynamic programming, the equivalent to the one stated in (4.4) and (4.5) for the appropriate value R, is:

$$\max_{m_{ij}} \prod_{i,j\in U} \left\{ \sum_{r=0}^{m_{ij}} \frac{(\lambda_{ij}t)^r}{r!} e^{-\lambda_{ij}t} \right\}, \tag{4.6}$$

$$\text{subject to: } \sum_{j=1}^{L} \sum_{i=1}^{N_j} w_{ij} (m_{ij}+\delta_{ij}) \leq W, \tag{4.7}$$

where W is a sufficiently large value of the weight of the spacecraft to allow attainment of the reliability R by a feasible selection of the variables, $m_{ij}$ and $\delta_{ij}$.

In the form of the notation found in [48], the dynamic programming formulation is presented by the following set of recursive relationships. $W_k$ represents a value of weight, the state variable, which is the input to stage k. The recursive relationships are written for the first few stages only, since the pattern then becomes obvious and further repetition is unnecessary to see the

general procedure. The stages are numbered in Table 4.2.

Stage 1:

$$f_1(W_1,m_{11}) = \max_{m_{11}} \left\{ \sum_{i=0}^{m_{11}} \frac{(\lambda_{11}t)^i}{i!} e^{-\lambda_{11}t} \right\} ,$$

from which it follows that $m_{11}^* = [(W_1-w_{11})/w_{11}]$, where $m_{11}^*$ is the optimum value of $m_{11}$, and the brackets indicate the largest integer value contained in the bracketed quantity.

Stages 2, 3 and 4:

$$f_k(W_k,m_{k1}) = \max_{m_{k1}} \left\{ \sum_{i=0}^{m_{k1}} \frac{(\lambda_{k1}t)^i}{i!} e^{-\lambda_{k1}t} \times f_{k-1}\left[W_k-(m_{k1}+1)w_{k1}\right] \right\} .$$

Stage 5:

$$f_5(W_5,m_{12},\delta_{12}) = \max_{m_{12},\delta_{12}} \left\{ \delta_{12} g_5\left(W_5,m_{12}\right) \times (1-\delta_{12}) f_4(W_4,m_{41}) \right\} ,$$

where

$$g_5\left(W_5,m_{12}\right) = \max_{m_{12}} \left\{ \sum_{i=0}^{m_{12}} \frac{(\lambda_{12}t)^i}{i!} e^{-\lambda_{12}t} \right\} ,$$

$m_{12}^* = [(W_5-w_{12})/w_{12}]$, and $W_4 = W_5$.

Note that in the above formulation it is implied that $\delta_{11} = \delta_{21} = \delta_{31} = \delta_{41} = 1-\delta_{12}$.

Stages 5, 6 and 7 are identical to stages 2, 3 and 4. Whenever a stage is reached which represents a module of the next higher level, the corresponding $\delta_{ij}$ variable is introduced in the same manner as with stage 5.

The above problem could also be solved using integer programming, however, the dynamic programming approach leads directly to an understanding of the nature of the commonality case.

Commonality case.

When the system includes several identical modules, each one being a candidate replacement module which is part of a different module at the next higher level, as illustrated in Figure 4.1, the straightforward dynamic programming solution is inadequate.

If the example system of Figure 4.1 is considered in the same manner as the no-commonality case of Figure 4.2, the procedure is followed in much the same way until the first component of path 2 is reached. It is obvious that prior decisions have been made as to the inclusion of an identical component to the one of the current stage, but with inadequate information. It is noted from the reliability constraint (4.2), that each term of the

product function includes several, $P_j$, terms each containing a decision variable, $\delta_{ij}^{(p)}$, associated with a different stage of the formulation. This characteristic, known as non-separability, prohibits the decomposition of the problem and thus the solution by dynamic programming.

The optimum module size problem will be formulated as an integer programming problem in the following section.

## Integer Programming Formulation

Given the hierarchy of candidate replacement modules described in detail at the beginning of this chapter and illustrated in Figure 4.1, the following expressions constitute an integer programming formulation of the optimum module size problem with common components. The notation is identical to that used previously to formulate the problem for the time independent case, except the failure probability, $P_{ij}$, is replaced by specifying the failure rate of module i at level j as $\lambda_{ij}$. The objective function for the integer programming formulation is unchanged from the previous statement, expression (4.1).

The constraints are divided into three sets. The first two sets insure that one and only one feasible set of modules, which are required for a single thread system, is selected. Constraint set one consists of a number of constraints equal to the number of paths, $P_1$, (values of p)

at the lowest modular level. The $\delta_{ij}^{(p)}$ corresponding to one of the modules of each path at the lowest level is summed with the $\delta_{ij}^{(p)}$ corresponding to the next higher level module within which the former is "contained". This sum is again added to the $\delta_{ij}^{(p)}$ corresponding to the next higher level module in the hierarchy which "contains" the modules of both of the previously summed $\delta_{ij}^{(p)}$. The sum is continued until the highest level module of the hierarchy is reached, and the value of the sum is required to be equal to 1.0. When the modules are numbered as in Figure 4.1, the first constraint of constraint set one can be written as:

$$\sum_{j=1}^{L} \delta_{1j}^{(1)} = 1. \qquad (4.8)$$

The remaining $(P_1-1)$ constraints of constraint set one are not written here in generalized form, since the notation required to do so is intractable. The example problem in Chapter V further clarifies the content of these constraints.

The second set of constraints insures that if one module of a given path at a given level is in the system, each of the modules at that level are also selected as modules of the basic system. The general form of the constraints of the second set is:

$$\delta_{i1}^{(p)} = \delta_{i+1,1}^{(p)} \quad ; \quad i = 1,\ldots, n_{pi} \ , \quad p = 1,\ldots, P_1, \tag{4.9}$$

where $n_{pj}$ is the number of different modules in path p at level j. There will be

$$\sum_{p=1}^{P_1} n_{p1} - P_1$$

such constraints in the set.

To complete the formulation, the third constraint set consists of only the single reliability constraint. Using the derivation of expressions (3.7) and (3.8) as a basis, the reliability constraint is:

$$\prod_{j=1}^{L} \prod_{i=1}^{N_j} \left[ \sum_{r=0}^{m_{ij}} \frac{(K_{ij}^{(p)} \lambda_{ij} t)^r}{r!} e^{-K_{ij}^{(p)} \lambda_{ij} t} \right] \geq R, \tag{4.10}$$

where $K_{ij}^{(p)} = \sum_{p=1}^{P_j} \delta_{ij}^{(p)} k_{ij}^{(p)}$ .

Except for this reliability constraint the problem consists of all linear functions with the variables required to be integers. It is noted that not only is constraint (4.10) nonlinear, but it is also non-separable. That is, in each of the

$$\sum_{j=1}^{L} N_j$$

product terms of the constraint, each of the variables $\delta_{ij}^{(p)}$, for which the module (i,j) is an element of path P, appears. The number of such variables appearing in each product term may be viewed as the degree of non-separability of the constraint. In general, all of the $(P_j)$ possible variables $\delta_{ij}^{(p)}$ could appear in each of the product terms. This would indicate that each module at level j is contained in each module at level j+1.

For computing purposes, a term of the constraint (4.10) for which $K_{ij}^{(p)} = 0$ is defined to be equal to 1.0.

Since the above integer programming formulation contains the non-separable reliability constraint, it appears to be amenable to practical solution only by the branch and bound approach discussed in the literature review. The next chapter will discuss the algorithm developed to solve the problem as formulated above.

## Data Requirements

The first step toward being able to analyze a spacecraft system to determine the optimum module size is to determine the feasible hierarchy of candidate modules as illustrated in Figure 4.1. The feasibility considerations, which are to be determined by design engineers, include such items as the practicality of providing accessibility, disconnect devices, storage, and the replacement skills involved at each modular level. Other considerations are the functional proximity of parts, and the packaging requirements. The major difficulty in this phase of analysis is likely to be the reduction of possibilities of modular configurations to a few cases selected for the analysis.

Once the modular levels to be considered have been determined, the weight and reliability of each of these candidate modules must be estimated. The factors to be considered in estimating these parameters were outlined under the Assumptions and Considerations section of this chapter. Of course, all of the problems of making estimates of design parameters for advanced spacecraft are attendant to this phase of obtaining useful data for the model.

It was stated earlier that the man-hours required to

replace a failed module with a spare is converted to weight. A suitable conversion factor reflects the increase in spacecraft weight required for providing a "man-hour of crew time". It is recognized that the resulting number of crew members required for replacement action must be rounded off to an integer value. It is expected that the crew maintenance time will be integrated with the duties of the basic spacecraft crew and that this factor will be resolved by determining an appropriate overall crew size.

# CHAPTER V

# A SOLUTION PROCEDURE FOR THE OPTIMUM MODULE SIZE

In the formulation of the optimum module size problem for the case involving identical components, the inability to obtain a solution by dynamic programming was mentioned. This chapter includes further discussion of the characteristics of the problem, a theoretical development and statement of a solution algorithm, and a discussion of variants of the algorithm. An example problem is stated, solved, and a discussion of sensitivity analysis presented. Reference should be made to Chapter IV as required for definition of terms and problem formulation.

## Discussion of the Solution Approach

There are two types of variables over which the objective is to be optimized: (1) $\delta_{ij}^{(p)}$, which is a binary (0 or 1) variable representing the decision concerning the inclusion of a given module in the basic system, and (2) $m_{ij}$, which takes on non-negative integer values representing the number of spares of the $i^{th}$ module at the $j^{th}$ level. For each feasible solution set of the $\delta_{ij}^{(p)}$ variable there remains to be solved the optimum redundancy problem for a single thread system with spares as discussed in

Chapter III. Each of the feasible sets of values of the $\delta_{ij}^{(p)}$ can be represented as a terminal node of a solution tree consisting of all feasible values of the $\delta_{ij}^{(p)}$ variables. This solution tree is illustrated in Figure 5.1 for the example problem of Figure 4.1. The values of the $\delta_{ij}^{(p)}$ variables for j=1 are not shown since their values are implied by assigning values to the variables for j=2 and 3. The left branch following each node represents the value of the variable equal to 0 and the right branch the value 1. Since there are 23 binary variables, there are $2^{23}$ solution sets for these variables, of which 45 are feasible. The 45 feasible solution sets are represented by the terminal nodes of the solution tree.

Each of the feasible solution sets can be analyzed as a candidate system design, and the minimum system weight to achieve a required reliability using spares can be determined. Once the $\delta_{ij}^{(p)}$ are assigned values, the reliability constraint becomes separable, containing only the $m_{ij}$ variables which represent the number of spares of each module, and the dynamic programming approach becomes feasible. The difficulty of solving the optimum spares redundancy problem for each feasible solution set of the $\delta_{ij}^{(p)}$ arises from the large number of these solution sets. In a problem of only slightly larger numbers of modules at each level than in the example problem of Figure 4.1, it

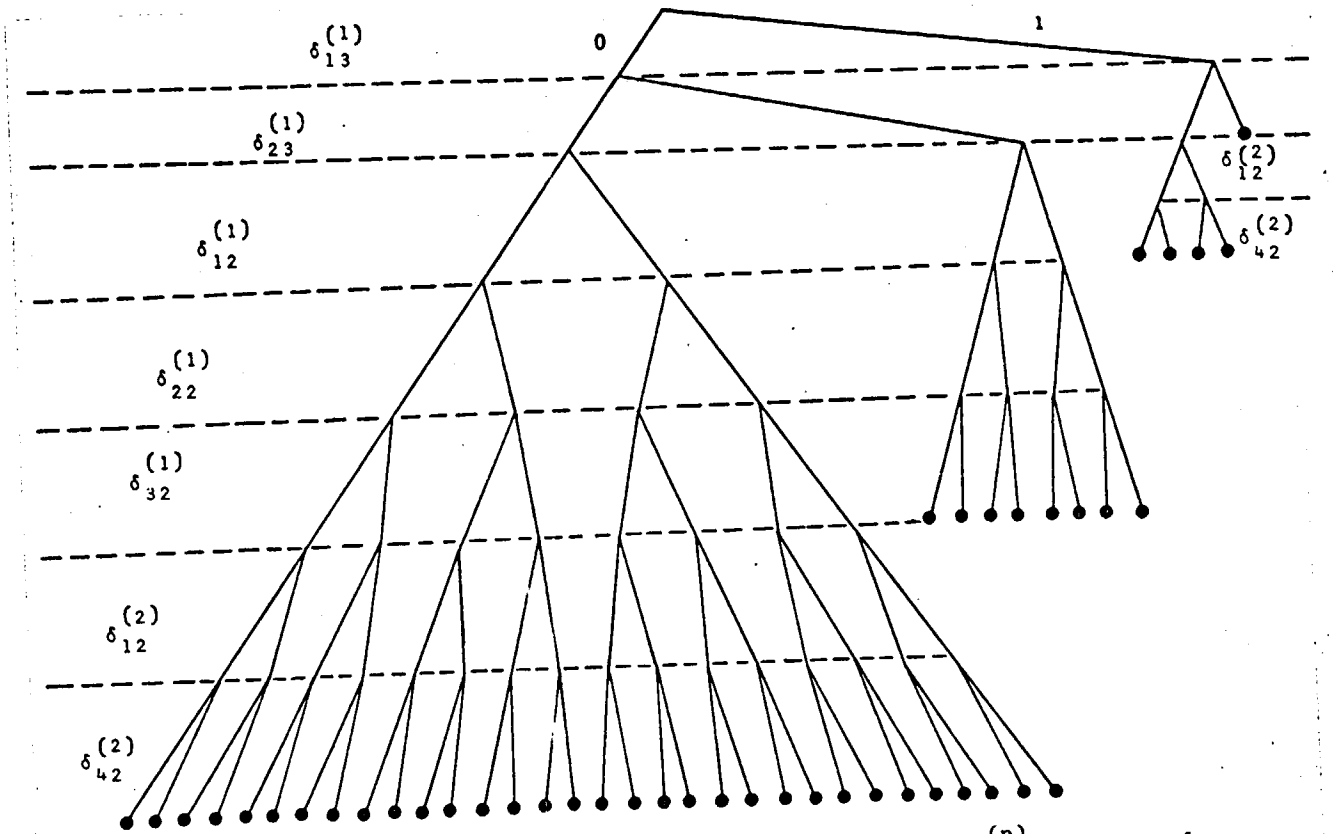


Figure 5.1.--Solution tree for feasible values of $\delta_{ij}^{(p)}$ for example problem of Figure 4.1.

becomes computationally infeasible to solve the number of dynamic programming problems which result. Hence, the concept of eliminating some of the branches of Figure 5.1 is suggested as a means of reducing the number of solution sets of the $\delta_{ij}^{(p)}$ variables which must be considered.

As discussed under the section on branch and bound methods in the literature review, one means of reducing the portion of the solution tree which must be explicitly considered is to bound the solutions in some of the branches and utilize these bounds as criteria for determining which solution sets must be evaluated. The algorithm stated in the following section employs two tests for eliminating branches of the solution tree.

The first of the tests imposed on branches of the solution tree of the $\delta_{ij}^{(p)}$ variables is an attempt to eliminate modules of the succeeding higher level from consideration. Consider the first two levels of modules, that is, for j=1 and j=2. For any path, p, of the lowest level, either (1) the modules of level 1, path p, ($\delta_{ij}^{(p)}=1$, i=1, ..., $n_{p1}$), (2) the next higher level module (j=2) in which all of the level 1 modules are contained, or (3) some higher level module must be in the system. If it can be shown that for the same weight the modules of a given path with corresponding spares have a higher reliability than the next higher module in which they are

contained with its spares, then the higher level module can be eliminated from consideration.

Figure 5.2 is a plot of the reliability versus weight of a hypothetical module at level 2, say, with spares as indicated by the increased weight (solid line). The dotted line represents the maximum reliability which can be obtained within the corresponding weight by adding spares to the level 1 modules contained within the level 2 module. Since one of the basic assumptions was that the reliability is always increased and the weight always decreased when a set of modules are aggregated to comprise a higher level module, before spares have been added the level 2 module line (solid) always lies above the level 1 line (dotted). It is only necessary to show that the level 1 line lies above the higher level line in the region of interest, that is, for reliabilities which the modules and their spares may have in the final system design. Since the system reliability is the product of the component set reliabilities, a lower bound on the component set reliabilities which must be examined is the required system reliability, R(R=.95 in Figure 5.2).

In Figure 5.2 the lower level module reliability is greater than the higher level module reliability in the region of interest. This condition will be called "dominance" of the component set. In the example problem

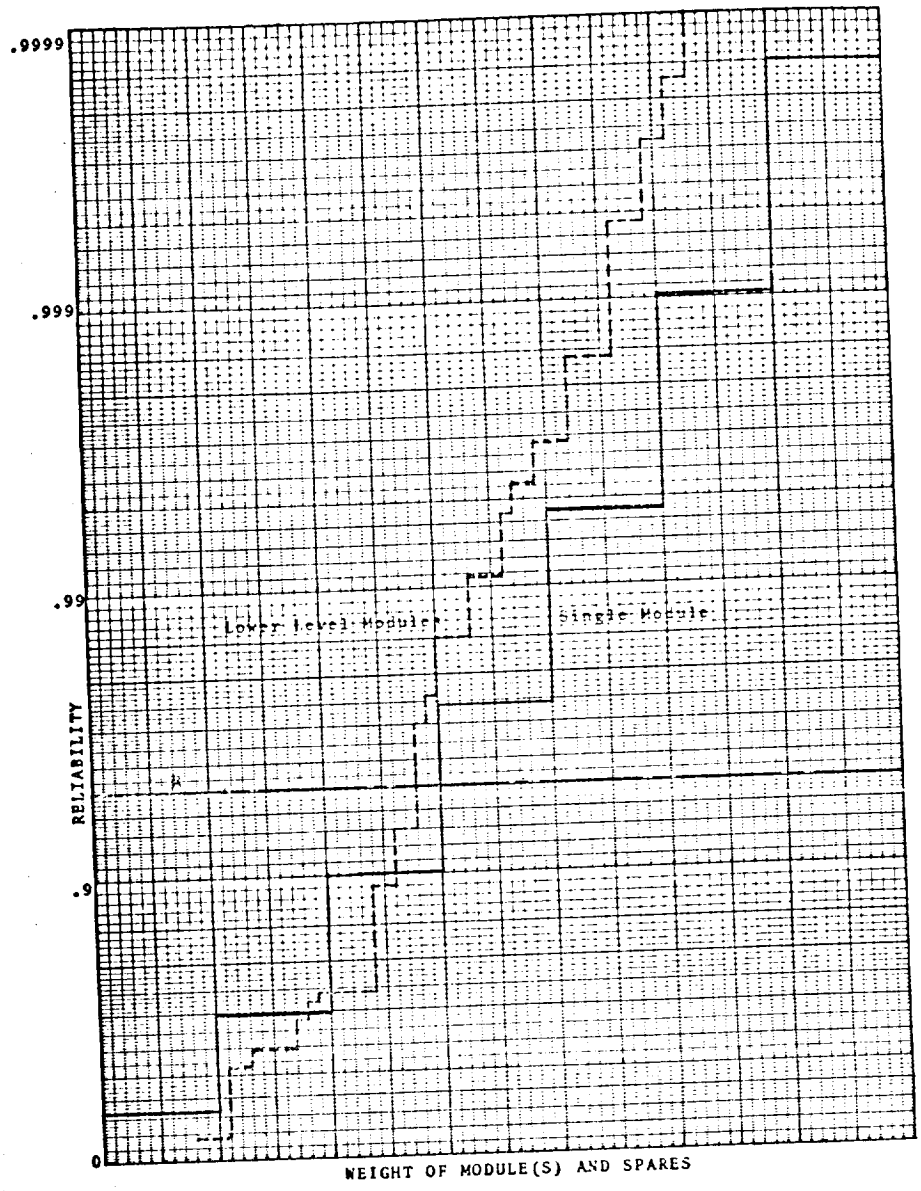


Figure 5.2.--Reliability vs weight for a module and its constituent components.

to follow later in the chapter, similar plots using the example problem will also show component sets which do not dominate the higher level module. If the higher level module line lies above the lower level line, the converse of the above situation, dominance does not result. This fact will be seen from the following arguments and an example of this case also appears in the example problem.

The second test used on branches of the solution tree of the $\delta_{ij}^{(p)}$ variables is again initiated at level 1. First the optimum solution to the dual problem (maximizing weight subject to requiring the reliability of the system to be greater than R) is solved for the system of all modules at level 1. Next, the solutions are obtained for each possible system of all modules at level 1 except for those of one of the $(P_1)$ paths, substituting in their place the corresponding level 2 module. If none of the systems with the level 2 module included has a lower weight than the level 1 system, the attempt to exclude some of the branches has failed, and the test is repeated with two level 2 modules simultaneously in the system.

If any of the solutions with a level 2 module included has a lower weight than the level 1 system, the module in level 2 corresponding to the system with the lowest weight dominates its constituent components, and they are eliminated from further consideration. By this test, the

branch of the solution tree representing the components of the level 2 module is excluded by being dominated by the lower bound of the solution having the level 2 module as an element. The lower bound in this case is a perfect lower bound; it is the optimal solution having the assigned subset of values of the $\delta_{ij}^{(p)}$ variable.

The system with the level 2 module is next considered as the level 1 system was previously, and the test is repeated for each of the remaining level 2 modules. At any point that the test fails, two modules of the next higher level are simultaneously considered, and the same tests repeated. The test proceeds with increased numbers of modules of the next higher level until all branches of the tree have been exhausted.

In the algorithm in the following section, optimization of several subproblems (trial solutions) is required. The method used for optimizing these subproblems is dynamic programming. The following dynamic programming formulations are referenced for the subproblem optimizations.

For step 1, the dynamic programming recursive relationship for each of the $P_1$ subproblems can be written:

$$f_n(W_{p1}, m_{n1}) = \max_{m_{n1}} \left\{ \sum_{r=1}^{m_{n1}} \frac{(k_{n1}^{(p)} \lambda_{n1} t)^r}{r!} e^{-k_{n1}^{(p)} \lambda_{n1} t} \times f_{n-1}(W_{p1} - m_{n1} w_{n1}) \right\}, \quad n=2,3,\ldots, n_{p1}, \tag{5.1}$$

and for n = 1,

$$f_1(W_{p1}, m_{11}) = \max_{m_{11}} \left\{ \sum_{r=1}^{m_{11}} \frac{(k_{11}^{(p)} \lambda_{11} t)^r}{r!} e^{-k_{11}^{(p)} \lambda_{11} t} \right\}.$$

For steps 7 and 9 the recursive relationship is of the form:

$$f_n(W, m_{nj}) = \max_{m_{nj}} \left\{ \sum_{r=0}^{m_{nj}} \frac{(K_{nj}^{(p)} \lambda_{nj} t)^r}{r!} e^{-K_{nj} \lambda_{nj} t} \times f_{n-1}(W - m_{nj}, w_{nj}) \right\}, \quad n=2,3,\ldots, n_{pj}, \tag{5.2}$$

and for n = 1,

$$f_1(W, m_{1j}) = \max_{m_{1j}} \left\{ \sum_{r=0}^{m_{1j}} \frac{(K_{1j}^{(p)} \lambda_{1j} t)^r}{r!} e^{-K_{1j} \lambda_{1j} t} \right\},$$

where

$$K_{nj}^{(p)} = \sum_{p=1}^{P_j} \delta_{nj}^{(p)} k_{nj}^{(p)}, \quad n=1,2,\ldots, n_{pj}.$$

The dual of the dynamic programming problem represented by equations (4.1) and (4.10) is also required in the algorithm, and using the previously defined notation, can be written:

$$\max_{m_{ij},\delta_{ij}^{(p)}} Z(m_{ij},\delta_{ij}^{(p)}) = \prod_{j=1}^{L}\prod_{i=1}^{N_j}\sum_{r=0}^{m_{ij}}\frac{(K_{ij}^{(p)}\lambda_{ij}t)^r}{r!}e^{-K_{ij}^{(p)}\lambda_{ij}t},$$

where $$K_{ij}^{(p)} = \sum_{p=1}^{P_j}\delta_{ij}^{(p)}k_{ij}^{(p)}, \qquad (5.3)$$

subject to: $$\sum_{j=1}^{L}\sum_{i=1}^{N_j} w_{ij}\left(m_{ij} + \sum_{p=1}^{P_j}\delta_{ij}^{(p)}k_{ij}^{(p)}\right) \leq W,$$

where W is a sufficiently large value of weight to allow attainment of the specified reliability, R.

## A Specific Algorithm

The following algorithm is presented as a particularization of the approach discussed in the previous section. The steps are to be followed consecutively unless noted otherwise.

1. Solve the following subproblems for $p=1,2,\ldots, P_1$ (a dynamic programming solution is given by expressions (5.1) in the previous section):

$$\max_{m_{i1}} r_{p1} = \prod_{i=1}^{n_{p1}} \sum_{x=0}^{m_{i1}} \frac{(k_{i2}^{(p)} k_{i1}^{(p)} \lambda_{i1} t)^x}{x!} e^{k_{i2}^{(p)} k_{i1}^{(p)} \lambda_{i1} t},$$

where modules $(i,1,p)$ are components of module $(i,2,p)$,

subject to: $$\sum_{i=1}^{n_{p1}} w_{i1} k_{i1} \leq W_{p1},$$

where $W_{p1}=1,2,\ldots, W_T$, and $W_T = \sum_{i=1}^{n_{p1}} (M_{i1} + 1) w_{i1}$.

2. Compute for $i=1, 2, \ldots, N_2$, $p=1, 2, \ldots, P_2$:

$$r_{i2}^{(p)} = \sum_{x=0}^{m_{i2}} \frac{(k_{i2}^{(p)} \lambda_{i2} t)^x}{x!} e^{-k_{i2}^{(p)} \lambda_{i2} t}, \quad m_{i2}=0,1,2,\ldots,M_{i2}.$$

3. For each value of $p=1,2,\ldots, P_1$, compare the

values of $r^*_{p1} = \max_{m_{i1}}\left\{r_{p1}\right\}$

with $r^{(p)}_{i2}$ for the level 2 module which can substitute for the level 1 modules; where for each comparison,

$$W_{p1} = u^{(p)}_{i2} ,$$

where $u^{(p)}_{i2} = 0,1,\ldots, U^{(p)}_{i2}$, and $U^{(p)}_{i2} = (m_{i2} + k^{(p)}_{i2})w_{i2}$.

4. If in step 3 for some p, for all comparisons such that $r^*_{p1} > R$ and $r^{(p)}_{i2} > R$,

$$r^*_{p1} \geq r^{(p)}_{i2} ,$$

where $r_{p2}$ corresponds to module (i,2):

a. If there are no other modules of type (i,2), go to 4c, otherwise:

b. Repeat steps 1-3 with $K_{i2} = \sum_{p=1}^{P_j} k^{(p)}_{i2}$ substituted for $k^{(p)}_{i2}$.

c. Set $\delta^{(p)*}_{i2} = 0$, where $\delta^{(p)*}_{ij}$ is the value of $\delta^{(p)}_{ij}$ for the optimum solution. If $\delta^{(p)*}_{ij} = 0$, all i, p, j=2,..., L, terminate. Solution is $\delta^{(p)}_{i1} = 1$, all i, p.

5. If in step 4 for some p=p', $\delta^{(p')}_{i2} = 0$ for all i, go to step 3 with p=p' and substitute "level 3" for level 2. Repeat this step for all levels.

6. For all i, p such that $\delta^{(p)}_{ij} \neq 0$ (by step 4) repeat steps 1 through 5 for j incremented by 1 in each step, j=1,2,..., L.

7. Solve the subproblems represented by equations (4.1) and (4.10). (The dual formulation (5.3) with a dynamic programming solution (5.2) is presented in the previous section).

where:

$$\text{a.}\quad \delta_{ij}^{(p)} = \begin{cases} 1, & j=1, \\ 0, & \text{otherwise}, \end{cases}$$

$$\text{b.}\quad \delta_{ij}^{(p)} = \begin{cases} 1, & j=1,\ p\neq p', \\ 1, & j=2,\ i=i'',\ p=p'', \\ 0, & \text{otherwise}, \end{cases}$$

($\delta_{ij}^{(p)}$ not previously set equal to 0),

where p' prepresents the path corresponding to the level 2 module (i'', p'') which can substitute for it (a dynamic programming solution is given in the previous section). Repeat for p'=1,2,..., $P_1$, with corresponding i'', p''. Designate the optimum values of Z (weight) in problem 7a, (7b), as $Z_a^*$, ($Z_b^*$).

8. If $Z_a^* \leq Z_b^*$ for all i, p, repeat step 7b, setting $\delta_{i2}^{(p)} = 1$ for all combinations of one additional number of values of i, i=1,2,..., $N_2$.

Otherwise, set $\delta_{i1}^{*(p')} = 0$, for all i,

for the solution $Z_b^{**}$, where

$$Z_b^{**}(m_{ij}, \delta_{il}^{*(p')}, \delta_{ij}^{(p)}) = \min_{m_{ij}, \delta_{ij}^{(p)}} Z_b^{*}(m_{ij}, \delta_{ij}^{(p)}) .$$

9. Solve the subproblems represented by equations (4.1) and (4.10) (see equations (5.2) and (5.3)), where:

$$\delta_{ij}^{(p)} = \begin{cases} 1, & j=1, p \neq p', p \neq p''', \\ 1, & j=2, i=i'', i=i'^{v}, p=p'', p=p'^{v}, \\ 0, & \text{otherwise}, \end{cases}$$

($\delta_{ij}^{(p)}$ not previously set equal to 0),

where p', p''' represent the paths corresponding to the level 2 modules (i'',p''), (i'$^{v}$,p'$^{v}$), respectively, which can substitute for them. Repeat for p'''=1,2,..., $P_1$, p'$\neq$p''', with corresponding i'',p'',i'$^{v}$,p'$^{v}$. Designate the optimum value of Z (weight) as $Z_9^{*}$.

10. If $Z_b^{**} < Z_9^{*}$ for all i, p, repeat step 9, setting $\delta_{i2}^{(p)}$ for all combinations of one additional number of values of i, i=1,2,..., $N_2$.

Otherwise, set $\delta_{il}^{*(p''')} = 0$, for all i,

for the solution $Z_9^{**}$, where

$$Z_9^{**}(m_{ij}, \delta_{il}^{*(p''')}, \delta_{ij}^{(p)}) = \min_{m_{ij}, \delta_{ij}^{(p)}} \left\{ Z_9^{*}(m_{ij}, \delta_{ij}^{(p)}) \right\} .$$

11. Repeat steps 9 and 10, incrementing by 1 the number of values of $\delta_{ij}^{(p)}$ set equal to 1, until all

$$\sum_{p=1}^{P_2} n_{p2}$$

values of $\delta_{ij}^{(p)}$ are assigned a value of 1 in a trial solution.

12. Assign $\delta_{i3}^{(p)} = 1$ ($\delta_{i3}^{(p)}$ not previously set equal to 0), for some i, p. Return to step 7 with level 3 substituted for level 2 in 7b. Repeat for each i, p.

13. Repeat step 12 for all combinations of one additional assigned $\delta_{i3}^{(p)} = 1$, i=1,2,..., $N_3$, p=1,2,..., $P_3$.

14. Return to step 12 with j=j+1, for all j=4,5,..., L.

15. The optimum solution is:

$$\min \left\{ Z_a^*, Z_b^{**}, Z_9^{**} \right\},$$

computed at each iteration of the algorithm.

## Theoretical Development

In steps 1-6 of the algorithm, a procedure is presented whereby certain branches of the solution tree (Figure 5.1) are determined to be dominated by another branch. The dominated branch represents the solutions for which a module of the next higher level than a particular set of components of the same path is in the system.

If there are no other modules of the type dominated by its constituent components (condition 4a), the consideration of the rest of the system clearly supports the assignment of $\delta_{i2}^{(p)*} = 0$ for the conditions stated in step 4. If, when additional paths are considered, other modules of the type dominated are encountered, the algorithm tests for dominance for the maximum number of modules which can be considered, $K_{i2}$. In this way if the addition of common modules causes the lines of Figure 5.2 to converge, even though with $k_{i2}^{(p)}$ modules dominance of the level 1 modules had occurred, the limiting case is tested before dominance is claimed and 4c executed.

Steps 7 through 14 of the algorithm describe a procedure whereby components which are dominated by the next higher level module in the hierarchy are excluded. The test for dominance of a higher order module over its components in steps 7 and 8 is accomplished by considering

first only a single module of level 2 with all other modules at level 1. In this way the advantage of commonality, discussed in Chapter III, are provided for the level 1 components to the largest extent. That is, any other combination of modules of level 1 and 2 would have less than or equal to the degree of commonality at level 1 than that used in the test. Hence, if a set of level 1 components is excluded from consideration in step 8 by setting $\delta_{i1}^{(p')^*} = 0$, the optimum solution cannot be improved with any of those $\delta_{i1}^{(p')^*} = 1$.

If a set of level 1 components is not dominated in steps 7 and 8, an enumeration of all combinations of two modules at level 2 is carried out. Thus exclusion of a possibly optimal set of values of $\delta_{ij}^{(p)}$ is avoided. If dominance of a higher level module over its components does not occur for any set of components, enumeration of the remaining branches of the solution tree of $\delta_{ij}^{(p)}$ variables is accomplished by the algorithm.

Each lower bound of a set of trial solutions is tested against the best solution previously obtained (incumbent) so that the optimal solution is at hand upon completion of the algorithm.

## Variants of the Algorithm

The algorithm presented in the previous section is only one of many which could be developed for the integer programming formulation. Depending upon the characteristics of the particular system being analyzed, one procedure may be more efficient than another.

One classification of systems to be considered is one with a large number of variables, that is candidate modules. Some of the modifications to the algorithm which may prove advantageous for this case are the following.

In steps 7 and 9, a lower bound on the particular branch of the solution tree specified was obtained by using dynamic programming, which results in a perfect lower bound (the optimum of the particular branch). Three types of alternatives are:

1. Use a technique other than dynamic programming to achieve the optimum completions.
2. Bound by relaxing the integer constraints.
3. Use an approximation technique for bounding.

The first of these three alternatives may be accomplished by linearizing the objective function (reliability) and using an integer programming algorithm to solve the resulting problem. Linearization can be accomplished by evaluating the terms of the objective

function (separated by taking logarithms) at all possible values of $m_{ij}$. Consecutive differences between the values of the objective at each integer value of $m_{ij}$ become the coefficients of binary (0 or 1) variables $m'_{ijk}$, where

$$m_{ij} = \sum_{k=0}^{M_{ij}} m'_{ijk} ,$$

and $M_{ij}$ is the maximum number of spares allowable for module i at level j. This general approach is explained in detail in [60].

The second alternative is to bound by relaxing the integer requirements. This method can be accomplished by linearizing the reliability expression as in alternative 1, and solving as a linear program, ignoring the integer requirement of the $m'_{ijk}$ variables defined above. The solutions obtained by this method (as a bound on the completion of a particular branch) have a higher value of the objective function (maximum reliability) than the true optimum completion using integer values.

The procedure for using linear programming as a bound is the same as the algorithm stated with the following exceptions. Step 7a is performed as given in the algorithm and the resulting trial solution serves as a comparative value against which to test the bounds obtained using

linear programming. Step 7b is performed using linear programming as described above. In step 8, if $Z_a^* \leq Z_b^*$ for all i, p, the remaining procedure is followed as stated in the algorithm. For those i, p for which $Z_a^* > Z_b^*$, repeat steps 7b and 8 as stated in the algorithm, finding $Z_b^{**}$ by dynamic programming. Next, step 9 is performed using linear programming. If $Z_b^{**} < Z_9^*$ for all i, p, the remaining procedure is followed as stated in the algorithm. For i, p such that $Z_b^{**} > Z_9^*$, repeat steps 9 and 10 using dynamic programming. Step 11 is performed using linear programming first, and, as in the above procedure, is repeated using dynamic programming for those i, p which are not excluded from consideration by the test using the linear programming bound.

It should be noted that the above procedure required re-solving (by an integer method) some of the trial solutions obtained using linear programming. The efficiency of the procedure depends upon the increased speed of computing the simplified linear program as well as the number of computations which must be repeated.

Alternative (3), using an approximation technique for bounding, is likely to be the most efficient method to solve the problem for very large systems. This alternative is accomplished by following the algorithm as stated except for steps 1, 7 and 9. In each of these steps, the

required solutions to the subproblems are obtained by approximation instead of by dynamic programming. It is recommended that the delta method be used for each solution. This method was explained in Chapter II, Redundancy Optimization Problems, and is also detailed in [52]. It should be noted that although the exact optimum is not guaranteed by alternative (3), in many cases the approximations will be identical to the exact optima, and useful solutions should generally result.

The algorithm was described in terms of a single thread basic system. Other configurations for the basic system can be considered. Equation (3.27) presented an expression for the reliability of a series-parallel system having two components in parallel. In order to determine the optimum module size for this system, it is necessary to substitute the objective function (3.27) for the objective (5.3) used in the algorithm. For the parameter N in (3.27) substitute $K_{ij}^{(p)}$, the number of components in the basic system for a component set i, j, p, where

$$K_{ij}^{(p)} = \sum_{p=1}^{P_j} 2\delta_{ij}^{(p)} k_{ij}^{(p)} .$$

Similarly, in the constraint (5.3), $\delta_{ij}^{(p)}$ must be multiplied by 2 to account for the weight of both parallel components

in each location. As indicated earlier, the computing time for equation (3.27) is significant. Its use even for moderate size problems may require use of one of the three alternatives listed above, especially alternative (3).

Another modification which can be made to the reliability expression used in the algorithm is to provide for components to have a different failure rate in each location in the basic system. This is easily accomplished by replacing equation (3.8) with (3.7) in formulation (5.3). No other modifications to the algorithm are required for this generalization.

## Sensitivity Analysis

The calculations inherent in the algorithm can be readily adapted for use in sensitivity analysis. First, the initial exclusion of higher level modules due to dominance by constituent lower level ones was performed for all values of reliability within the range of interest. Hence, any change of the required system reliability within the range of interest will not invalidate the results of these dominance tests.

Secondly, each candidate branch of the solution tree (of binary variables representing modules of the single thread system) was bounded using dynamic programming. Hence, the bound for any value of reliability is at hand

for each of the solutions. Therefore, it is not necessary to re-calculate the dynamic programming tests for bounds for the first set of next-higher-level components (steps 7 and 9), since the solution for them is available. Only in the case where an additional attempt at higher level module bounding did not fail in the original problem and failed in the second, must additional dynamic programming bounding be accomplished.

The effects of a change in the system reliability requirement is evaluated by repeating steps 8, 10, 11, 12, and 13 in the algorithm, assuming the dynamic programming output tables have been retained. The sensitivity of the solution to changes in spacecraft weight is analyzed in the same manner.

If the delta method is used for alternative 3 in the preceding section, the solution is computed for all values of reliability less than or equal to R, the required value. This allows evaluation of the effect on system weight of a change in reliability. However, this should not be confused with the information provided by a dynamic programming solution. With dynamic programming several alternate modules may enter into the solution for slight changes in R. With the delta method, reducing R merely

results in a solution obtained by removing the spares from the system in reverse order from that in which they were added.

If alternative 1 in the preceding section is used, a sensitivity analysis will be much more difficult to accomplish. By using integer programming, the advantage of having a solution for the range of the state variable (weight) is relinquished. In this case a change in the reliability requirement or weight constraint requires re-computing the algorithm.

It has been shown in Chapter IV that the number of spares required for a set of components is quite sensitive to whether or not the components are identical. Similar analyses can be performed for the entire system, providing more realistic results, by successive computation of the algorithm for a system with and without assumed commonalities.

## An Example Problem

The example system illustrated in Figure 4.1 will be used to demonstrate the solution procedure. Table 5.1 contains the set of values for the parameters (weight and failure rate) used for the example problem. The value of the required reliability is 0.80.

| i \ j | 1 | | 2 | | 3 | |
|---|---|---|---|---|---|---|
| | wt. | λt | wt. | λt | wt. | λt |
| 1 | 8 | .1 | 18 | .5 | 50 | 2.0 |
| 2 | 6 | .2 | 10 | .8 | 25 | .6 |
| 3 | 2 | .3 | 12 | .45 | | |
| 4 | 1 | .35 | 11 | .25 | | |
| 5 | 3 | .05 | | | | |

TABLE 5.1 VALUES OF PARAMETERS FOR EXAMPLE PROBLEM

Figure 5.3 illustrates the solution tree for the $\delta_{ij}^{(p)}$ variables for the example problem. The solutions are numbered for reference purposes. The heavy lines represent the solutions which are enumerated. The remaining branches are excluded, or implicitly enumerated, by the algorithm.

Step 1. The values obtained for the dynamic programming solution to the $P_1=5$ subproblems are represented by the dotted lines in Figures A.1 to A.5 (in Appendix).

Step 2. The values for the computations required by this step are also plotted in Figures A.1 to A.5, represented by the solid lines.

Step 3. Comparison of $r_{p1}^{*}$ and $r_{p2}$ can be made directly from Figures A.1 to A.5.

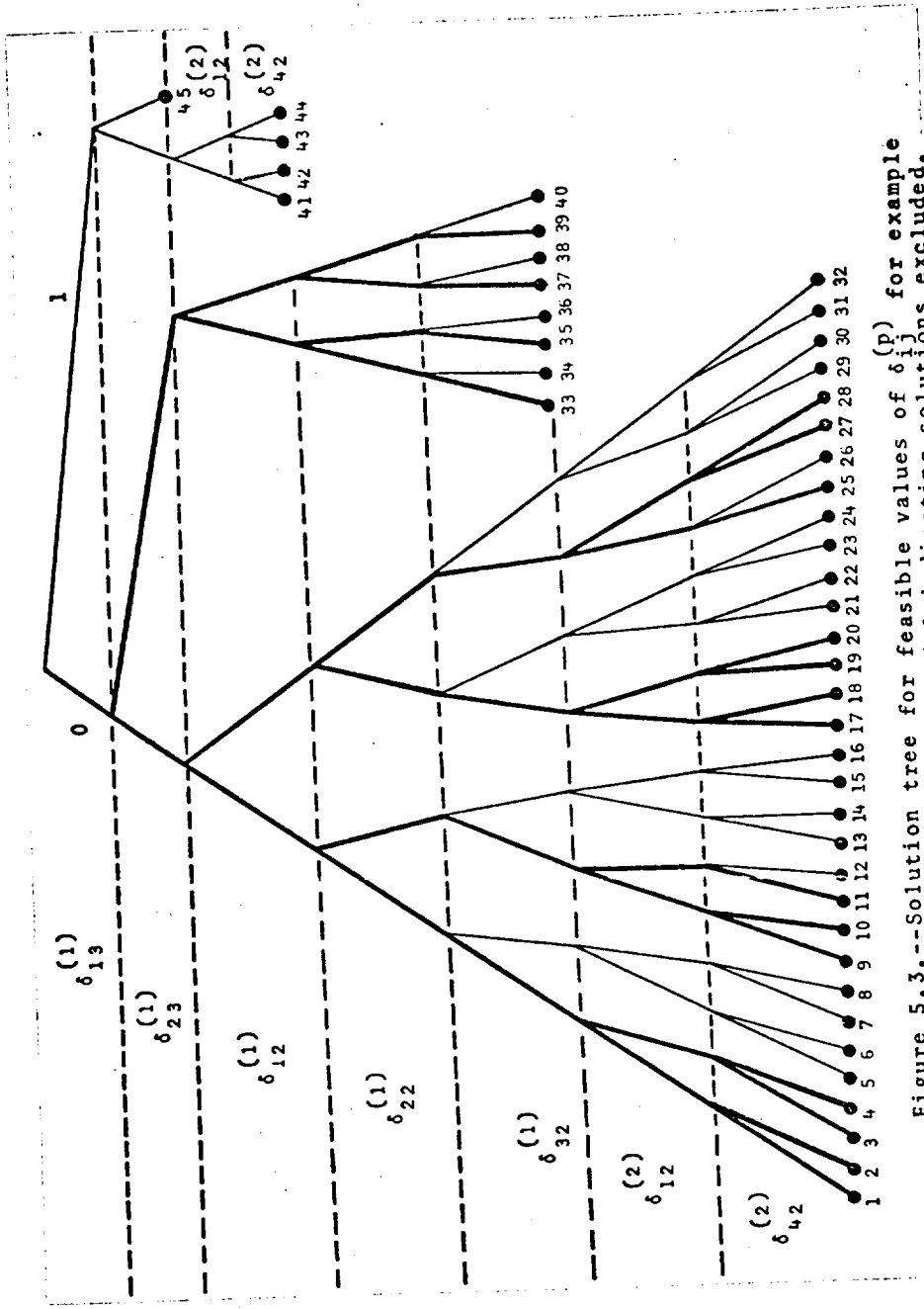


Figure 5.3.--Solution tree for feasible values of $\delta^{(p)}_{ij}$ for example problem of Figure 4.1 indicating solutions excluded.

Step 4. $r^*_{p1} \geq r^p_{i2}$ , $r^*_{p1} > .80$ and $r^p_{i2} > .80$

for $r^*_{p1} = r^*_{31}$ , $r^p_{i2} = r^1_{32}$ .

Set $\delta^{(1)*}_{32} = 0$. The following solutions in Figure 5.2 are eliminated by this step: 5,6,7,8,13,14,15,16,21,22,23,24, 29,30,31,32,34,36,38,40.

Step 5. Go to step 6, since no p=p' exists such that $\delta^{(p')}_{i2} = 0$ for all i.

Step 6. Set j=2 and return to step 1.

Step $1^1$.† The values obtained for the dynamic programming solution to the $P_2$=2 subproblems are represented by the dotted lines in Figures A.6 and A.7.

Step $2^1$. The values for the computations required by this step are also plotted in Figures A.6 and A.7, represented by the solid lines.

Step $3^1$. Comparisons of $r^*_{p2}$ and $r_{p3}$ are made directly from Figures A.6 and A.7.

Step $4^1$. $r^*_{p2} \geq r^p_{i3}$ , $r^*_{p2} > .80$ and $r^p_{i3} > .80$

for $r^*_{p2} = r^*_{12}$ , $r^p_{i3} = r^1_{13}$ .

Set $\delta^{(1)*}_{13} = 0$. Solutions 41-45 are eliminated from consideration by this step.

---

†The step number superscripted by k indicates the $k^{th}$ repetition of the step.

Step 5'. Go to step 6, since no p=p' exists such that $\delta_{i2}^{(p)} = 0$ for all i.

Step 6'. j has been incremented to L-1=2. Go to step 7.

Step 7. The optimum value of weight for the subproblems are:

a. $Z_a^* = 251$

b.

| p' | 1 | 2 | 4 | 5 |
|---|---|---|---|---|
| (i'',p'') | (1,1) | (2,1) | (1,2) | (4,2) |
| $Z_b^*$ | 253 | 253 | 257 | 253 |

The above solutions are the values for solutions 1, 2, 3, 9, and 17 in Figure 5.3.

Step 8. Since $Z_a^* < Z_b^*$ for all i'',p'', step 7b is repeated for all combinations of two modules, (i'',p'').

Step 7'. The optimum values for the required subproblems are (j=2):

| p' | 1,2 | 1,4 | 1,5 |
|---|---|---|---|
| i'',p'' | (1,1),(2,1) | (1,1),(1,2) | (1,1),(4,2) |
| $Z_b^*$ | 255 | 241 | 257 |

| p' | 2,4 | 2,5 | 4,5 |
|---|---|---|---|
| i'',p'' | (2,1),(1,2) | (2,1),(4,2) | (1,2),(4,2) |
| $Z_b^*$ | 264 | 257 | 262 |

The above solutions are the values for solutions 4, 10,11,18,19,25 in Figure 5.3.

Step $8^1$.

$$Z_b^{**} = 241$$

Set $\delta^*_{i1}(p') = \delta^{(1)}_{11} = \delta^{(1)}_{21} = \delta^{(1)}_{31} = \delta^{(1)}_{51} = \delta^{(4)}_{11} = \delta^{(4)}_{21} = \delta^{(4)}_{31} = \delta^{(4)}_{51} = 0.$

The solutions of Figure 5.3 which are eliminated from consideration by this step are 12 and 26.

Step 9. The optimum values for the required subproblems are (p' determined in step $7^1$):

| p',p''' | (1,4), 2 | (1,4), 5 |
|---|---|---|
| [(i'',p'')],(i'',p'') | [(1,1),(1,2)],(2,1) | [(1,1),(1,2)],(4,2) |
| $Z_9^*$ | 242 | 243 |

The above solutions are the values for solutions 20 and 27 in Figure 5.3.

Step 10. $Z_b^{**} = 241 < Z_9^*$ for all i,p. Repeat step 9 for all combinations of 3 values of i, $(\delta^{(p)}_{i2} \neq 0)$.

Step $9^1$. The optimum value for the subproblem required is (only one remains):

| p',p''' | (1,4), 2,5 |
|---|---|
| [(i'',p')];(i'',p'') | [(1,1),(1,2)];(2,1),(4,2) |
| $Z_9^*$ | 241 |

$Z_9^*$ is the value for solution 28 in Figure 5.3.

Step $10^1$. $Z_b^{**} = 241 = Z_9^*$ Set $\delta_{i1}^{(p''')} = \delta_{11}^{(2)} = \delta_{21}^{(2)} = \delta_{31}^{(2)} = \delta_{11}^{(5)} = \delta_{21}^{(5)} = \delta_{51}^{(5)} = 0$ or carry $Z_b^{**} = Z_9^* = 241$ as an alternative solution.

Step 11. All

$$\sum_{p=1}^{P_2} n_{p2}$$

values of $\delta_{ij}^{(p)}$ have been assigned a value of 1 in a trial solution.

Step 12. Assign $\delta_{23}^{(1)} = 1$ ($\delta_{13}^{(1)}$ previously set equal to 0).

Step $7^2$. The optimum value of weight for the sub-problems are:

a. $Z_a^* = 254$ (Solution 33 in Figure 5.3)

b.

| p' | 1 | 2 |
|---|---|---|
| i'',p'' | (1,1) | (2,1) |
| $Z_b^*$ | 256 | 262 |

The $Z_b^*$ correspond to solutions 35 and 37 in Figure 5.3.

Step $8^2$. $Z_a^* < Z_b^*$. Repeat step 7b for $\delta_{i2}^{(p)} = 0$ for $i = (1,2)$.

Step $7^3$.

b.

| | |
|---|---|
| p' | 1,2 |
| i'',p'' | (1,1),(2,1) |
| $Z_b^*$ | 251 |

($Z_b^*$ corresponds to solution 39 in Figure 5.3).

Step $8^3$. $Z_a^* > Z_b^*$. Set $\delta_{i1}^{(1)} = 0$, i = 1,2,3,5, $\delta_{i1}^{(2)} = 0$, i = 1,2,3, (for $\delta_{23}^{(1)} = 1$).

Steps 9,10,11,12,13, and 14 are satisfied by having reached j=L or completing all required subproblems.

Step 15. The alternative optima are $Z_b^{**}$ for step $7^1$ and $Z_9^*$ for step $9^1$, having a value of 241, represented by solutions 19 and 28, respectively in Figure 5.3.

The values of the $\delta_{ij}^{(p)}$ variables for each optimum solution are:

$$Z_b^{**}: \quad \delta_{ij}^{(p)} = \begin{cases} 1, & \text{all } (i,1,p) \neq (1,1,1),(1,1,2) \\ 1, & (i,j,p) = (1,2,1)(1,2,2) \\ 0, & \text{otherwise.} \end{cases}$$

$$Z_9^*: \quad \delta_{ij}^{(p)} = \begin{cases} 1, & \text{all } (i,2,p) \neq (3,2,1) \\ 1, & (i,j,p) = (1,1,3),(4,1,3) \\ 0, & \text{otherwise.} \end{cases}$$

The values of the $m_{ij}$ variables for each optimum solution are:

($m_{ij}$ = 0 for (i,j) not listed)

$Z_b^{**}$:

| (i,j) | (1,2) | (1,1) | (2,1) | (3,1) | (4,1) | (5,1) |
|---|---|---|---|---|---|---|
| $m_{ij}$ | 3 | 2 | 2 | 3 | 3 | 1 |

$Z_9^{*}$:

| i,j | (1,1) | (4,1) | (1,2) | (2,2) | (4,2) |
|---|---|---|---|---|---|
| $m_{ij.}$ | 1 | 3 | 4 | 2 | 1 |

# CHAPTER VI

## SUMMARY AND CONCLUSIONS

The question of optimization of spacecraft configuration with regard to replacement module size has not been quantitatively treated in the literature. Therefore, this research has been exploratory, devoting considerable effort to the determination of the nature of the problem presented by this question, and leading to several new questions which are outlined in the succeeding chapter.

The entire subject of maintainable spacecraft systems is relatively unexplored from an optimization viewpoint. Most of the literature cited discusses spacecraft maintenance as though the basic hardware design to be maintained were already determined. It is generally agreed that some degree of maintenance will be desirable for extended manned space missions, but what level of maintenance activity should be performed remains an open question. The degree of maintenance considered in this research was confined to replacement of failed components with spares.

Until the present time manned spacecraft have not been designed for in-space maintenance. The spacecraft designers have been concerned with optimization of each component of the spacecraft with regard to the performance of its required function. One effect of this approach,

which results from designing for short duration flights, is that each component tends to be unique. When maintenance by modular replacement with spares is considered, the uniqueness of components becomes a severe detriment to the objective of minimizing total spacecraft weight.

In Chapter III, the effects of having identical components in several locations in the basic design of spared systems were indicated. Other effects resulting from the derivation of expressions for the reliability of spared systems were also demonstrated. One purpose of these presentations is to bring about an awareness of the design implications of maintenance by sparing. For example, it was illustrated in Figure 3.9 that a considerable increase in individual component weight in order to achieve component commonality can still result in a reduction of total spacecraft weight, due to the reduced spares requirement. The utilization of these "suboptimal" components for the purpose of achieving commonality should be thoroughly explored for optimal design of spacecraft for extended manned space missions.

An expression for the reliability of a series-parallel spared system of two parallel components and any number of components in series was developed. The reliability of this type of system has not appeared in the literature, except for the work [41], which was shown in Chapter II to

be incorrect. This configuration is useful in spacecraft systems to avoid any system downtime while a failed component is being replaced. The reliability of the series-parallel spared system is compared graphically with that of a series system with spares for various numbers of spares and failure rates.

It is demonstrated that addition of active parallel components in systems with spares result in less reliability increase than addition of the same number of spares, and may even reduce the system reliability. Therefore redundancy by active parallel components should only be used when components cannot be spared or when system downtime cannot be allowed while failed components are being replaced.

The optimum module size problem was formulated both with and without common components in the system. For the system without common components dynamic programming was used to solve for the optimum module size. It was shown that with commonality the optimum module size problem cannot be solved using dynamic programming because the reliability function is non-separable. An integer programming formulation was presented for this problem.

To solve the integer program for the optimum module size, it was necessary to use a branch and bound type approach. A rather involved algorithm was developed, which

includes the use of dynamic programming for excluding branches of the solution tree.

Several variants of the algorithm were discussed. The objective of the variants of the algorithm is to explore efficient ways of handling systems with a large number of alternative modules. The most promising of these alternatives is the use of an approximation technique such as the delta method for obtaining trial solutions. This alternative retains a "built in" sensitivity analysis in a limited sense, because the (approximate) optimum solution is computed for all values of reliability less than or equal to the required value.

Although example problems have been solved using the algorithm, a comprehensive computer program for the procedure has not been written. A computer program for each of the dynamic programming subproblems has been written, and it remains to incorporate this program into an integrated program.

The mathematical model and solution procedure which have been developed in this research appear to be adequate for solving the problem as presented within the limitations of the stated assumptions. The data required for exercise of the algorithm for actual spacecraft systems must still be obtained. It is recommended that a small segment of a system be used as a starting point from which to gain

experience with the determination of the type of data required. It is expected that as much benefit will be derived from the consideration of alternatives required as input to the algorithm as from actual use of the solution procedure to obtain a specific optimum spacecraft design.

# CHAPTER VII

## RECOMMENDATIONS FOR FUTURE RESEARCH

Since this research has approached a problem which has not been previously reported in the literature, there are a multitude of extensions and unanswered questions to be investigated. It is expected that the optimum module size problem in the context presented will provide a basis for extensive further study of spacecraft design for long duration manned space missions.

One of the guidelines set forth for this research was that bench repair is not allowed. An extension of this work would be to determine the reliability of the system with the capability to repair failed modules, and trade-off the provision of a repair capability with fewer spares versus the system without repair.

Similar to a repair capability, but requiring less elaborate equipment, is the provision for replacement of parts within failed modules. In this manner, failed modules may be salvaged and usable components reclaimed for use elsewhere. An analysis of this capability would be involved with the extent that this modified component accessibility affects the module weight and reliability.

Another realistic consideration for long space missions is to allow the switching of components used in

non-critical systems to critical locations when spares have been exhausted. Even within the same system, for example in Figure 2.3, it would be reasonable to allow using components from each of the N cells to replace failed components in other cells after all of the spares have been used. Under this policy the system would fail only when less than N components remain in operation.

A number of simplifying assumptions were used in deriving each of the reliability expressions in Chapter III. In order to make the model more realistic for its intended application, there are several of these which it would be desirable to change.

It was assumed that the replacement action for restoration of failed modules occurs in the following ideal manner: (1) instantaneously, and (2) causing neither the replacement module nor any other component to fail. The first of these assumptions is supported by the relatively short replacement times in comparison to the overall mission duration. However, an analysis of the effect of assumed replacement times on the spacecraft weight due to requiring parallel components instead of spares would be of considerable interest.

The second assumption, accounting for the reliability of the replacement action, could be a critical factor in the design of spacecraft for long duration missions. If

this factor is accounted for, it would be expected that the optimum module size would occur at a higher level. The greatest hindrance to the practicality of research in this area is obtaining estimates of the replacement action reliability, especially with regard to causing components other than the replacement module to fail. The simple approach of increasing the component failure rate to account for unreliable replacement action, although crude, may be adequate relative to the available data. It is recommended that some attempt be made to acquire this type of data before attempting to incorporate this factor into the model.

The assumption that there is no degradation of the reliability of spares, although not severely restricted or unrealistic, should warrant further study. The effect of including this consideration is primarily in requiring additional input data to the model and in making the reliability expressions algebraically cumbersome.

A subject of much discussion in the reliability literature, but one which is relatively unexplored for complex systems such as derived in this research, is that of considering a failure distribution other than exponential. The exponential distribution is reasonable for many space system components since the failure probability may be proportional to the time of exposure to

space hazards. However, for long space missions wear out of components becomes a factor in some cases, and research in complex systems with other failure distributions is needed.

In this research all modules in the spacecraft were assumed to be mission critical. If this assumption is changed to allow even two levels of criticality, there are many problems which must be resolved. First, the number of parallel components for each cell of Figure 2.3 would logically be greater for critical components than for the less critical. The development of the reliability for the resultant system would pose an interesting but involved problem. Secondly, under various criticality levels, ambiguity with regard to which modules to include in the system required to attain a given reliability, R, would result. In spite of these limitations, it is recommended that research in quantitative assessment of criticality levels be undertaken.

# APPENDIX

# RELIABILITY VERSUS WEIGHT CURVES

# FOR

# EXAMPLE PROBLEM

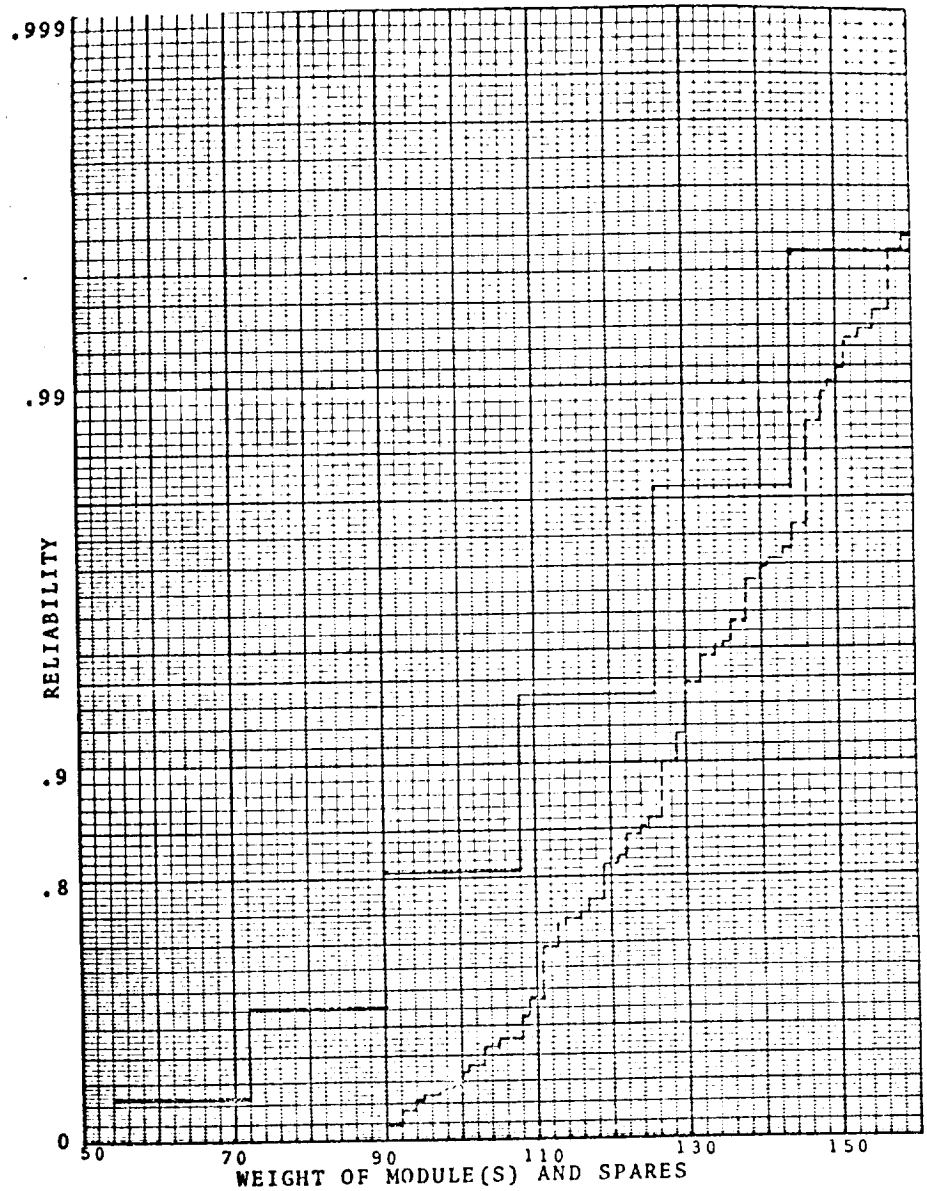


Figure A.1.--Reliability vs weight for module (i,j,p)= (1,2,1) and its constituent components.

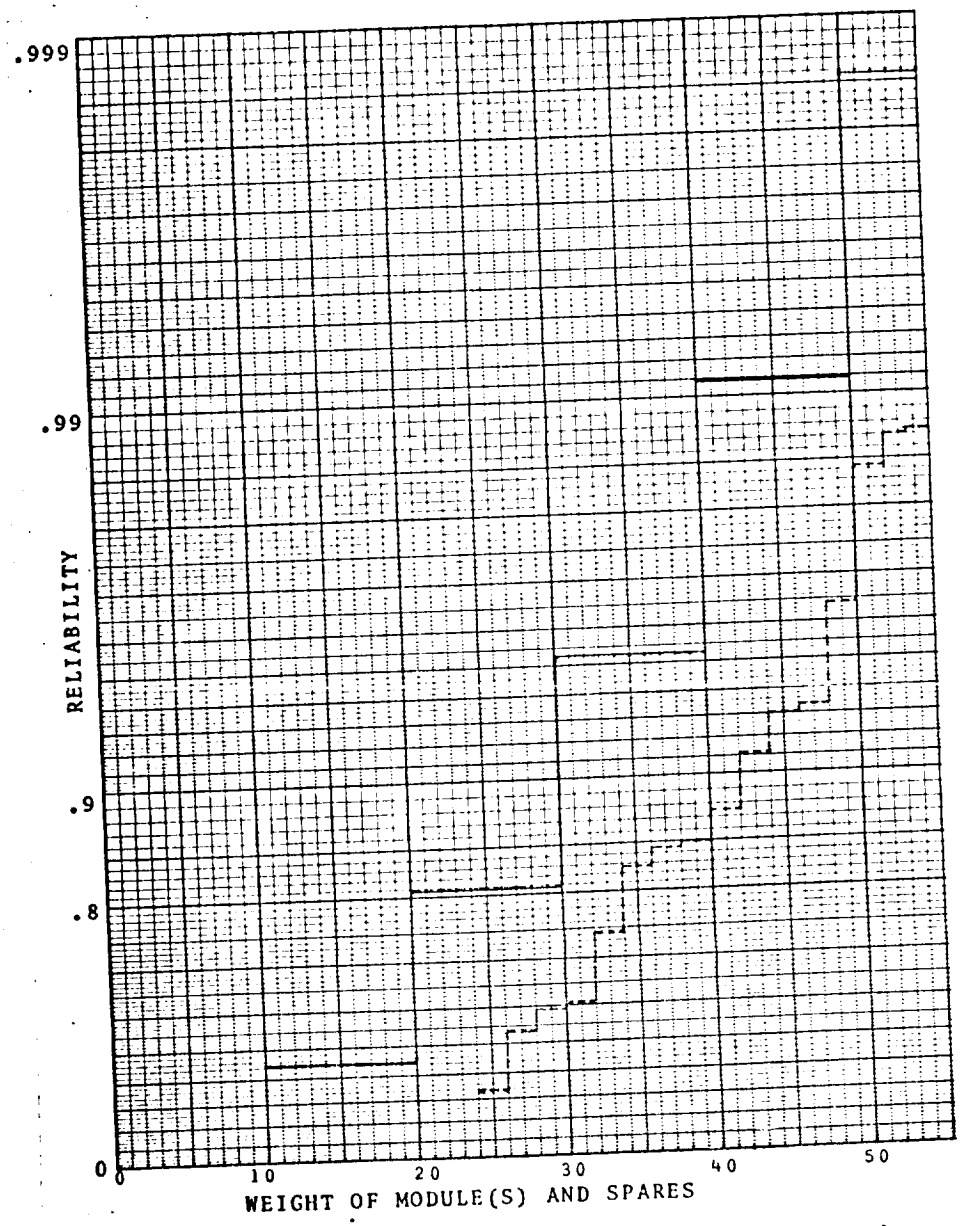


Figure A.2.--Reliability vs weight for module (i,j,p)= (2,2,1) and its constituent components.

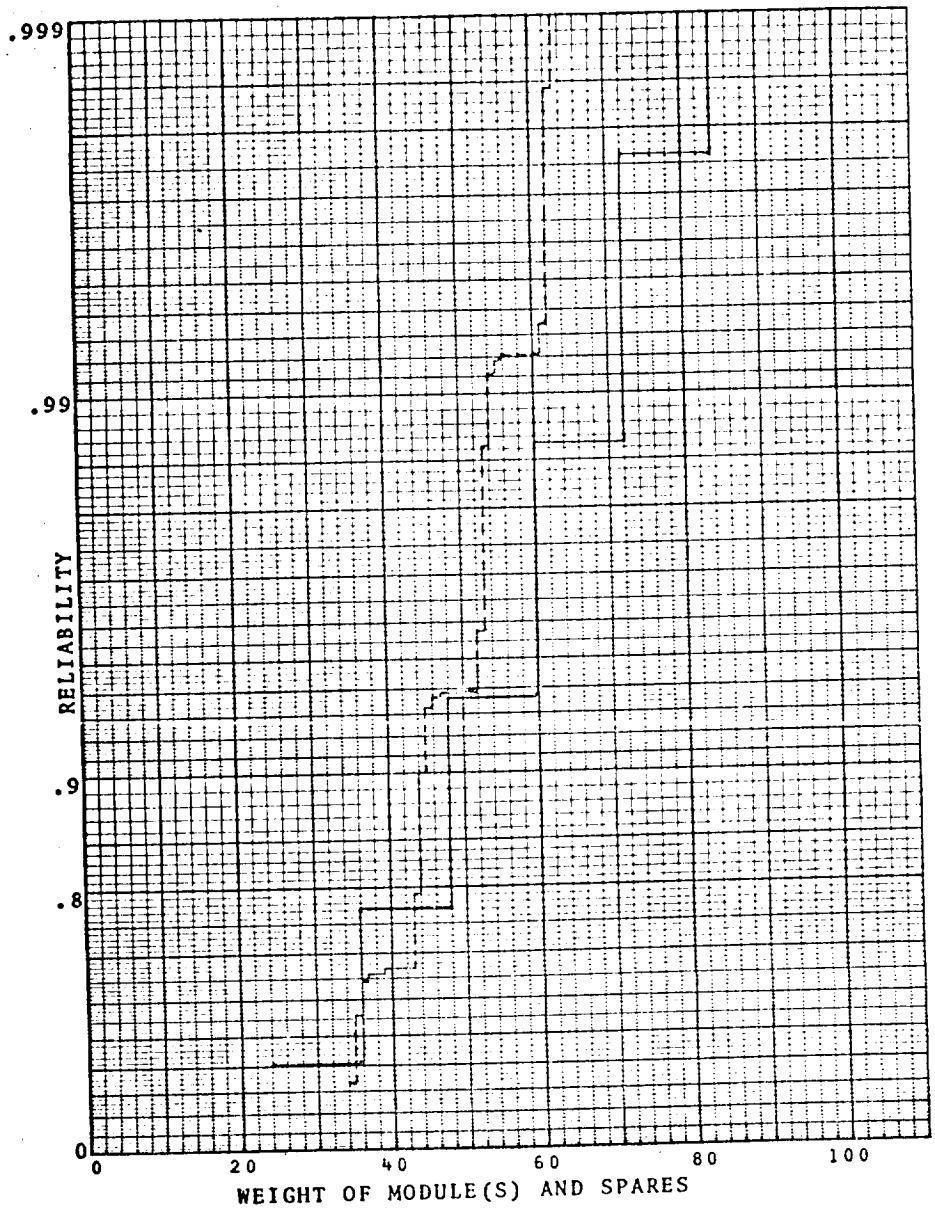


Figure A.3.--Reliability vs weight for module (i,j,p)= (3,2,1) and its constituent components.

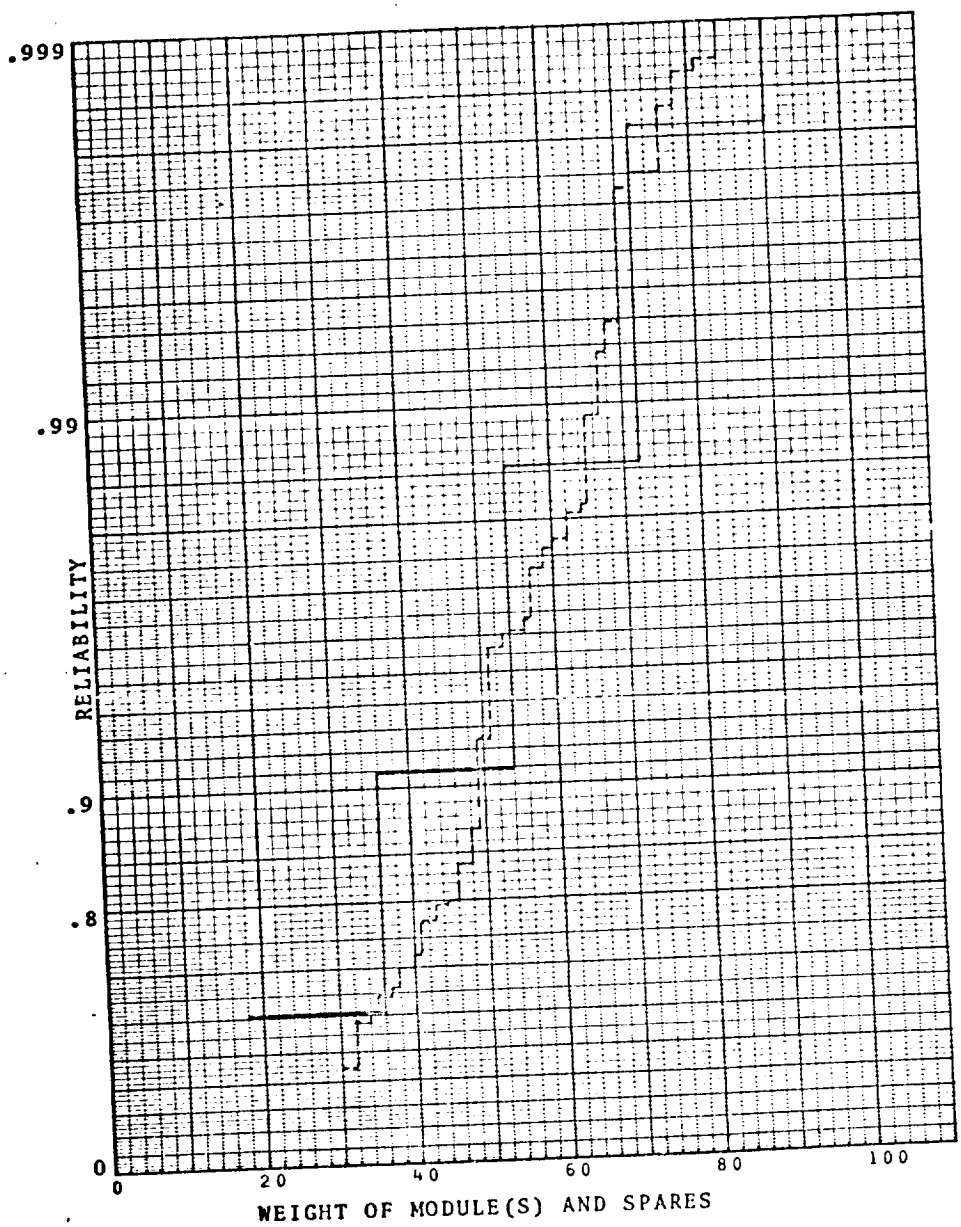


Figure A.4.--Reliability vs weight for module (i,j,p)= (1,2,2) and its constituent components.

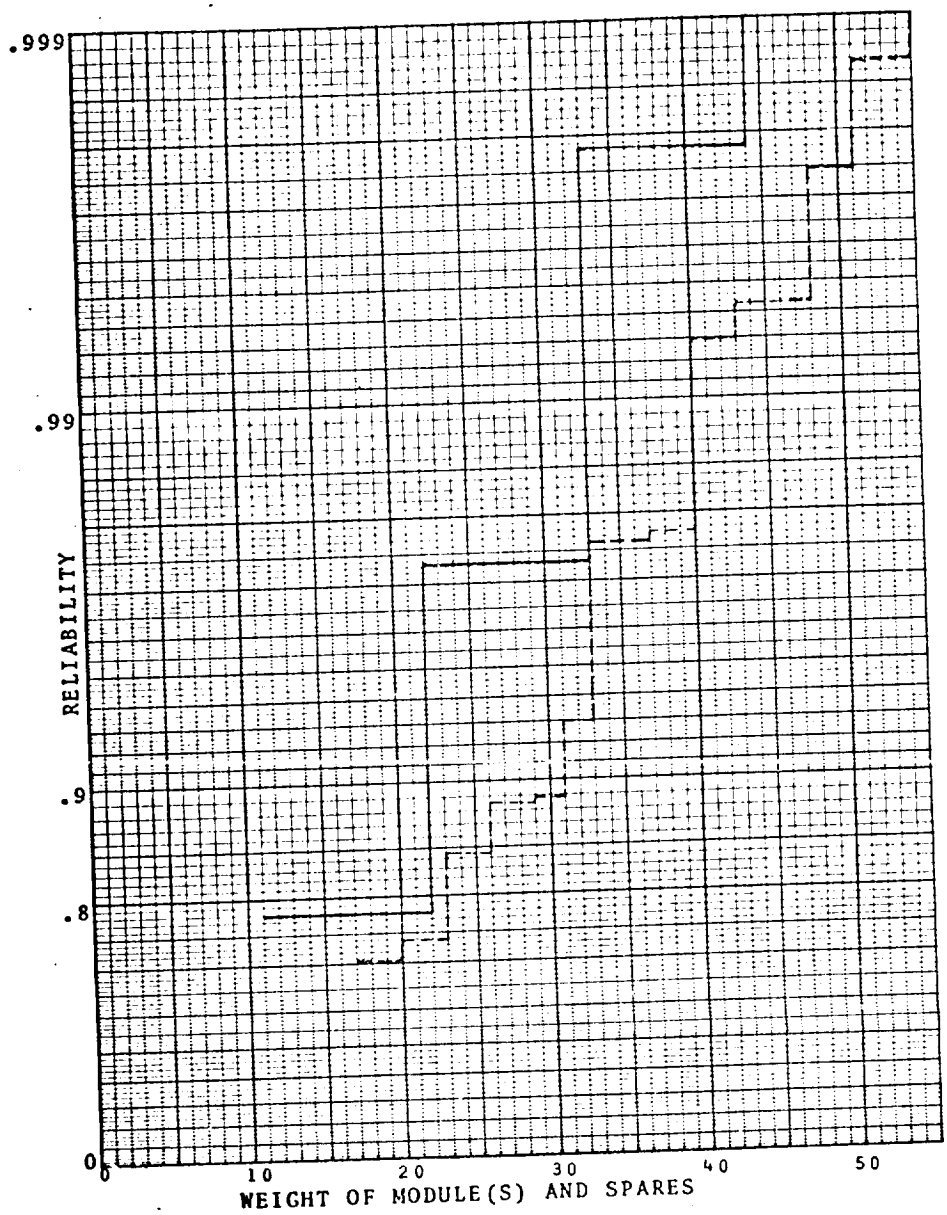


Figure A.5.--Reliability vs weight for module (i,j,p)= (4,2,2) and its constituent components.

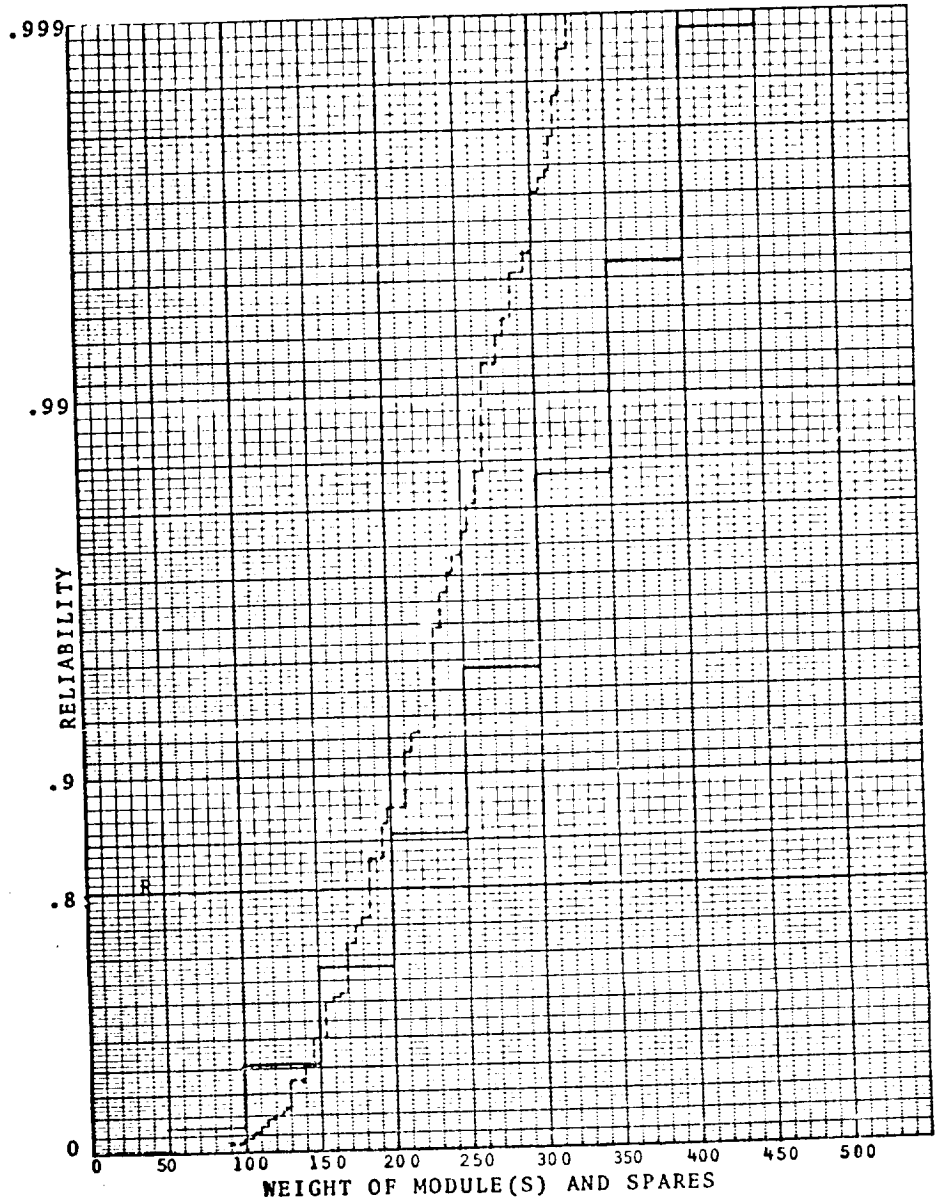


Figure A.6.--Reliability vs weight for module (i,j,p)= (1,3,1) and its constituent components.

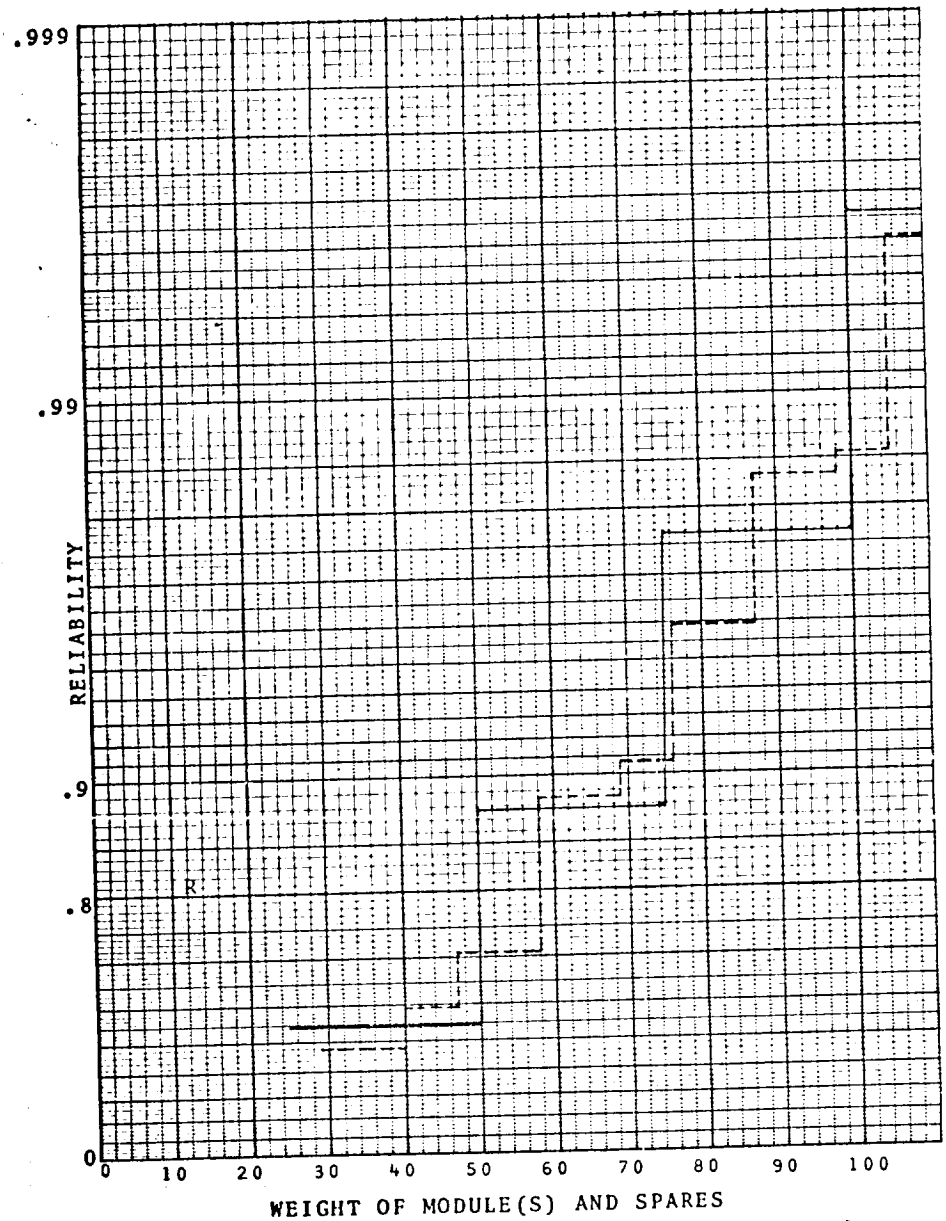


Figure A.7.--Reliability vs weight for module (i,j,p)= (2,3,1) and its constituent components.

## REFERENCES


1. Egon Balas, "An Additive Algorithm for Solving Programs with Zero-One Variables," Operations Research, Vol. 13, 517-545 (1965).

2. M. L. Balinsky, "Integer Programming: Methods, Uses, Computation," Management Science, Vol. 12, 253-313 (1965).

3. R. E. Barlow and F. Proschan, Mathematical Theory of Reliability, John Wiley & Sons, New York, 1965.

4. I. Bazovsky, Reliability Theory and Practice, Prentice-Hall, Englewood Cliffs, New Jersey, 1961.

5. R. E. Bellman and S. E. Dreyfus, Applied Dynamic Programming, Princeton University Press, Princeton, New Jersey, 1962.

6. _____ AND _____, "Dynamic Programming and the Reliability of Multicomponent Devices," Operations Research, Vol. 6, 200-206 (1959).

7. _____, Dynamic Programming, Princeton University Press, Princeton, New Jersey, 1957.

8. J. O. Black, "Maintenance Aspects in Extended Space Operations," SID65-1484. North American Aviation, October, 1965.

9. S. R. Calabro, Reliability Principles and Practices, McGraw-Hill Book Company, Inc., 1962.

10. Gary R. Carey, "Optimization of System Redundancy Under Multiple Constraints, "Unpublished Master's Thesis, Texas A&M University, January, 1967.

11. R. B. Carpenter, "A Reliability Concept for Long Space Missions", North American Aviation, Inc., Space and Information Systems Division, SID65-1255, October 12, 1965.

12. D. N. Chorafas, Statistical Processes and Reliability Engineering, D. Van Nostrand Company, Inc., 1960.

13. R. V. Churchhill, Modern Operational Mathematics in Engineering, McGraw-Hill Book Company, New York, 1944.

14. G. B. Dantzig, Linear Programming and Extensions, Princeton University Press, Princeton, New Jersey, 1963.

15. D. J. Davis, "Increased Reliability Through Spare Parts Optimization," Proceedings Aerospace Reliability and Maintainability Conference, 1964.

16. W. D. Day, Introduction to Laplace Transforms, Interscience Publishers, Inc., New York, 1960.

17. L. T. Fan, et al., "Optimization of Systems Reliability," IEEE Transactions on Reliability, Vol. R-16, September, 1967.

18. S. I. Firstman, "Current Concepts and Issues of In-Space Support," The RAND Corporation, P-2966, August, 1964.

19. B. J. Flehinger, "Reliability Improvement Through Redundancy at Various System Levels," IBM Journal of Research and Development, V. 2, No. 2, PP. 148-158 (1958).
Also IRE National Convention Record, part 6, 1958.

20. M. A. Geisler and H. W. Karr, "The Design of Military Supply Tables for Spare Parts," Operations Research, Vol. 4, 431-442 (1956).

21. A. M. Geoffrion, "Implicit Enumeration Using an Imbedded Linear Program," Working Paper No. 120, Western Management Science Institute, University of California, May, 1967.

22. _____, "Integer Programming by Implicit Enumeration and Balas' Method," The RAND Corporation, RM-4783-PR, February, 1966.

23. F. Glover, "A Multiphase-Dual Algorithm for the Zero-One Integer Programming Problem," Operations Research, Vol. 13, 879-919 (1965).

24. _____, "Truncated Enumeration Methods for Solving Pure and Mixed Integer Linear Programs," Working Paper 27, Operations Research Center, University of California, Berkeley, May, 1966.

25. B. V. Gnedenko, Y. K. Belyaev, and A. D. Solov'yev, Mathematical Methods of Reliability Theory,

Moscow, Nauka, Physics and Mathematics Literature Section, 1965.

26. A. S. Goldman and T. B. Slattery, Maintainability: A Major Element of System Effectiveness, John Wiley & Sons, New York, New York, 1964.

27. R. Gordon, "Optimum Component Redundancy for Maximum System Reliability," Operations Research, Vol. 5, 229-243 (1957).

28. M. H. Gourary, "A Simple Rule for the Consolidation of Allowance Lists," Naval Research Logistics Quarterly, Vol. 5, 1-15 (1958).

29. G. Hadley, Nonlinear and Dynamic Programming, Addison-Wesley Publishing Co., Inc., Reading, Mass., 1964.

30. R. V. Hees and H. W. Meerendonk, "Optimum Reliability of Parallel Multi-Component Systems," Operations Research Quarterly, Vol. 12, May, 1961.

31. A. E. Holmes, et al., "Integrated Maintainability Criteria Development Technique," Martin-Marietta Corporation, Technical Report AFAPL-TR-67-93, September, 1967.

32. W. D. Ineson, Reliability Handbook, McGraw-Hill Book Company, Inc., New York, 1966.

33. M. W. Joeseph, "The Reliability Dilemma in Space Program Packaging," Proceedings Aerospace Reliability and Maintainability Conference, 1964.

34. J. O. Kettelle, Jr., "Least-Cost Allocation of Reliability Investment," Operations Research, Vol. 10, 249-265 (1962).

35. P. J. Kolezar, "Linear Programming and the Reliability of Multicomponent Systems," Naval Research Logistics Quarterly, Vol. 14, 317-328 (1967).

36. A. H. Land and A. G. Doig, "An Automatic Method of Solving Discrete Programming Problems," Econometrics, Vol. 28, 497-520 (1960).

37. E. L. Lawler and M. D. Bell, "A Method for Solving Discrete Optimization Problems," Operations Research, Vol. 14, 1098-1112 (1966).

38. _____ and D. E. Wood, "Branch-and-Bound Methods: A Survey", Operations Research, Vol. 14, 699-719 (1966).

39. J. M. Liittschwager, "Dynamic Programming in the Solution of a Multistage Reliability Problem", Journal of Industrial Engineering, Vol. 15, No. 4, July-August (1964).

40. D. K. Lloyd and M. Lipow, Reliability: Management, Methods, and Mathematics, Prentice-Hall, Englewood Cliffs, New Jersey, 1962.

41. "Maintainability and Reliability Cost Effectiveness Program", D2-22022-7, The Boeing Company, May, 1965.

42. "Maintainability of Manned Spacecraft for Long-Duration Flights", Vol's. I, II, and III, Contract NAS 2-3705, The Boeing Company, Space Division, Seattle, Washington, July, 1967.

43. "Manned Mars and/or Venus Flyby Vehicle Systems Study", Contract NAS 9-3499, Reliability Analysis, Vol. 5, North American Aviation, Inc., June, 1965.

44. M. Mazumdar and A. J. Federowicz, "Use of Geometric Programming to Maximize the Reliability Achieved by Redundancy", Research Report 67-1C3-RECAP-R1, Westinghouse Research Laboratories, March 20, 1967.

45. D. F. Morrison, "The Optimum Allocation of Spare Components in Systems", Technometrics, Vol. 3, 399-406 (1961).

46. P. M. Morse, Queues, Inventories and Maintenance, John Wiley and Sons, Inc., New York, 1965.

47. R. H. Myers, et al. (ed.), Reliability Engineering for Electronic Systems, John Wiley & Sons, Inc., New York, 1964.

48. G. L. Nemhauser, Introduction to Dynamic Programming, John Wiley and Sons, Inc., New York, 1966.

49. B. G. Peters, D. Amorelli and J. T. Celentano, "Spacecraft Reliability + Man = Mission Success" Proceedings Aerospace Reliability and Maintainability Conference, 1964.

50. E. L. Peterson, "Maintainability in Future Space Systems," Advanced Logistics, Space and Information System Division, North American Aviation, Inc. (Company Report - No Date).

51. Erich Pieruschka, Principles of Reliability, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1963.

52. F. Proschan and G. Black, "On Optimal Redundancy," Operations Research, Vol. 7, 581-588 (1959).

53. _____ and T. A. Bray, "Optimum Redundancy Under Multiple Constraints," Operations Research, Vol. 13, 800-814 (1965).

54. G. E. Roberts and H. Kaufman, Table of Laplace Transforms, W. B. Saunders Company, Philadelphia, 1966.

55. W. H. Robertson, Tables of the Binomial Distribution Function for Small Values of P, Sandia Corporation, SCR-143, January, 1960.

56. G. H. Sandler, System Reliability Engineering, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1963.

57. B. C. Spradlin, "An Advantage of Sum Constraints for Dynamic Programming Algorithms," The Journal of Industrial Engineering, Vol. XVII, No. 9, September, 1966.

58. Tables of the Individual and Cumulative Terms of Poisson Distribution, Defense Systems Department, General Electronic Company, D. Van Nostrand Company, Inc., Princeton, New Jersey, 1962.

59. F. A. Tillman, "Integer Programming Solutions to Constrained Reliability Optimization Problems," Transactions of the Twentieth Annual Technical Conference, American Society for Quality Control, 1966.

60. _____ and J. M. Liittschwager, "Integer Programming Formulation of Constrained Reliability Problems," Management Science, Vol. 13, N887-899 (1967).

61. C. G. Vandervoort and H. S. Dordick, "Selection Between Redundancy and Repair in Manned Spacecraft: A Designer's Tool," The RAND Corporation, No. RM-4325-NASA, September, 1964.

62. _____, "The Effect of Maintainability on the Mission Reliability of Two-Element Redundant Spacecraft Subsystems," The RAND Corporation, RM-4824-PR, December, 1965.

63. W. H. Von Alvin, (ed.), Reliability Engineering, ARINC Research Corporation, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.

64. Dale White, "An Optimization Approach to the Solution of the Maintenance Float Problem," Unpublished Master's Problem, Texas A&M University, January, 1968.

65. C. T. Whitehead, "Selection of Spares and Redundancy for the Apollo Spacecraft," The RAND Corporation, RM-4177-NASA, August, 1964.

66. D. R. Wilken and E. S. Langford, "Failure Probability Formulas for Systems With Spares," Operations Research, Vol. 14, 731-733 (1966).

67. M. Zelen, Statistical Theory of Reliability, University of Wisconsin Press, Madison, Wisconsin, 1963.

# VITA

Donald E. Wagner was born at Port Arthur, Texas, September 19, 1934, son of Mr. and Mrs. Edward G. Wagner. He attended elementary and high school in Port Arthur, graduating from Thomas Jefferson High School in 1952.

While serving four years active duty in the U.S. Navy, he graduated from the Navy Electronics School at Treasure Island, California, and served on a destroyer from 1954-56.

Mr. Wagner received his B. S. in Industrial Engineering from Lamar State College of Technology, Beaumont, Texas in 1960. In 1960-61, he was a Walter P. Murphy Fellow at Northwestern University, Evanston, Illinois, receiving his M. S. in Industrial Engineering, Operations Research option. Before obtaining his M. S. he worked as an Industrial Engineer at Texaco, Inc., Port Arthur, Texas for two summers. From 1961-64 he was employed by Eastman Kodak Company at Rochester, New York, in the Industrial Engineering Division.

He is currently employed by the National Aeronautics and Space Administration, Manned Spacecraft Center, Houston, Texas, working in areas of Operations Research from 1964 to 1966, when selected for the Graduate Study Program, to begin doctoral work at Texas A&M University.

Address: 209 Bayou View Drive, Seabrook, Texas, 77586.

The typist for this dissertation was Mrs. Mary Bradley.

A Reproduced Copy

OF

Reproduced for NASA

*by the*

**NASA Scientific and Technical Information Facility**

FFNo 672 Aug 65